EXPERIENCE IS NOT EVIDENCE: COVERAGE-CALIBRATED MEMORY REUSE FOR TOOL-USING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tool-using agents increasingly reuse their own past experience, yet a memory that looks relevant can quietly damage the trajectory it was meant to help. Having experience is not the same as having evidence that it transfers. We therefore replace the retrieval objective with an interventional one: the transfer gain of a request–memory pair, measured by running the same request with and without that memory, and generalized across pairs by tool-call structure rather than text similarity. What may be claimed about this gain is determined by coverage. Where practice contains no comparable structure, the sign of transfer is unidentifiable, making forced use-or-discard decisions provably lossy and motivating a genuinely resolving third action. We accordingly distinguish an average-value guarantee over candidate sets from a marginal per-pair certificate. The average-value guarantee opens on fresh confirmatory practice data and, with the policy frozen in advance, on a fresh fixed-design grid over held-out deployment domains. In contrast, independent validation with the representation frozen produces no pair-level openings and defers throughout. On a controlled benchmark designed so that the harmful candidate is the more textually similar one, only structurally incompatible memories are reliably harmful. Similarity and an LLM judge fail to identify them, whereas transfer-based gating incurs no observed harmful use on any tested model family and matches oracle utility on two of the three full-benchmark executors. Reliable harm appears only in our authored competing memories. A preregistered organic pilot produced isolated negative outcomes that failed its prespecified reliability gate and did not survive replication, bounding the scope of the result and reinforcing the need to measure transfer rather than infer it from a memory label.

1 INTRODUCTION

Modern tool-using agents reuse experience they generate themselves: pre-task synthetic practice (Choi et al., 2026; Zhang et al., 2026b), self-reflective traces (Shinn et al., 2023), distilled experiential insights (Zhao et al., 2024), induced workflow memory (Wang et al., 2025), and episodic memories retrieved at inference (Park et al., 2023). A growing body of evidence shows this is double-edged, identifying several *distinct* failure modes: an *experience-following* bias, where high input similarity to a retrieved memory drives the agent to imitate it even when doing so is wrong (Xiong et al., 2026); memory-induced tool-drift (Dabas et al., 2026); forgetting under continual reuse (Hu et al., 2026); and risks from self-evolution (Shao et al., 2026). The common assumption is that a *relevant* or previously *successful* memory is safe to reuse. It is not.

From relevance to transfer; from experience to evidence. We replace the objective. The unit of analysis is the request–memory *pair*, and the quantity of interest is the *incremental* value of using that memory, that is, how much the full-trajectory outcome changes relative to the same agent run without it. This packages two claims. First, *relevance is not the objective*: usefulness is a property of the pair, measured by an incremental intervention effect, not of the memory in isolation. Second, *having experience is not having evidence that it transfers*: a positive predicted effect may be supported by ample nearby practice or almost none, and only in the former case may the agent act on it. The principle is that experience should be *vetted* for transfer before it is reused, and we

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

make it operational: predict transfer, certify it where coverage supplies evidence, abstain where it does not, and *acquire* the missing evidence.

What the claim is not. Not that wrong memories can hurt, nor that one should abstain when uncertain; both are unsurprising. It is that the safety-relevant object is the request-specific *incremental transfer effect* rather than memory quality or relevance, and that whether one may act on it is constrained first by *structural coverage* and, conditional on coverage, by finite-sample uncertainty. So a memory about a *different* object attains the same observed mean transfer as the right-object one when tool-call structure matches (both $+0.167$, against -0.247 for same-object wrong-action), and abstention is not a tunable margin: without coverage the sign is *unidentifiable* (Theorem 2).

Why coverage, but not the usual coverage. Off-policy theory uses importance-weight and concentrability coefficients to characterize when logged data support policy evaluation (Swaminathan & Joachims, 2015; Munos & Szepesvári, 2008; Chen & Jiang, 2019), but two things make the naive version unusable: the relevant coverage is over the *tool-call structure of the pair*, not raw request text (a supportive and a corrupted memory for the same request must map to different representations, or no gate can separate them); shift-aware conformal methods use importance weighting under covariate shift, or quantify coverage degradation under non-exchangeability (Tibshirani et al., 2019; Barber et al., 2023), and the coefficient depends on the unknown deployment law. We use a structural *pair* representation and a deployment-weighted proxy, treating residual representation and evaluator error as audited bias terms.

Contributions. Our starting point is an estimand: the paired full-trajectory contrast $\tau(x, m) = V^{\pi_m}(x) - V^{\pi_0}(x)$ produced by injecting one specific memory. Our central result is that this estimand is not always actionable, and that fact dictates the rest of the design. The two obstacles are hierarchical rather than competing: coverage decides whether a valid claim is *identifiable* at all, and conditional on coverage, finite-sample uncertainty decides whether it is *actionable*. Our frozen validation fails at the second stage, not the first.

1. **An impossibility result that motivates the decision structure.** Where practice supplies no coverage of a deployment structure, *no* method can uniformly match the oracle: every mandatory binary gate suffers regret $\geq \alpha\Delta/2$ (Theorem 2). The architecture is motivated by this lower bound under an oracle-regret objective, not derived as the only possibility.
2. **A resolving third action can be minimax-optimal.** If deferral *resolves* the current request at cost c_d, it is minimax-optimal whenever $c_d \leq \Delta/2$ (Corollary 2). Our deployed defer does not resolve anything: it falls back to no-memory and marks an acquisition need whose benefit accrues to *later* requests, so the corollary motivates the architecture rather than certifying the mechanism we run. Unlike learning to defer (Madras et al., 2018; Mozannar & Sontag, 2020), that third action requests *evidence* for a structural region rather than handing the decision to an expert.
3. **Unresolved coverage as a diagnostic acquisition principle.** Deferral localises *where* evidence is missing, and acquiring paired practice there is a monotone-submodular objective with the classical greedy guarantee on a frozen-round surrogate (§5). We present this as a principle rather than a demonstrated repair: in our runs acquisition tightens intervals without opening a per-pair certificate.
4. **Two guarantee semantics, kept apart.** What may be claimed must track coverage: a candidate-set average-value guarantee versus a marginal per-pair certificate, with $\lambda < 1$ never called certified (§4).
5. **An empirical separation between the two claim strengths.** On a benchmark where the harmful candidate is deliberately the *more* similar one, structurally incompatible memory is reliably harmful (-0.42 corrupted, -0.25 distractor). An average-value guarantee is attainable on both sides: confirmatory TRACE-AVG opens on a fresh 384-task practice fold, and the frozen policy then opens a bound of $+0.032$ on a fresh fixed-design grid of 384 held-out-domain tasks, with no transport term and 0 harmful uses of 128. Independently validated pair-level certification is not, at the evidence level we can afford: a frozen-representation validation fold of $46{,}080$ episodes certifies nothing ($0/32$ uses, $32/32$ deferrals). Average usefulness can be established with moderate evidence; pair-level usefulness may stay uncertifiable even after expensive validation.

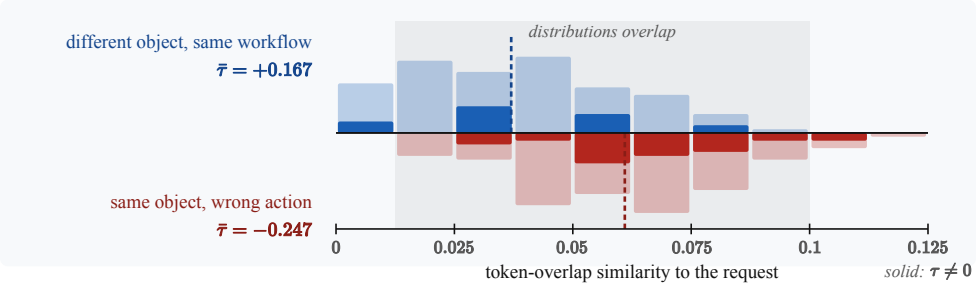


Figure 1: **Relevance does not determine transfer.** All 96 request–memory pairs per class, binned by token-overlap similarity and mirrored about a shared axis; darker bars count pairs whose measured gain was non-zero. The same-object, wrong-action class sits to the *right*, so it is the *more* similar (0.061 versus 0.037 mean, dashed), yet its non-zero pairs are all negative and the different-object, same-workflow pairs all positive. The distributions overlap across the shaded band, so no similarity threshold separates the classes.

Relation to prior work. The ingredients exist separately: effect-based memory utility (Srivastava, 2026; Guan et al., 2026), risk-sensitive memory control (Iscan, 2026), trajectory-level conformal risk (Opoku & Banahene, 2026), and classical off-policy evaluation and safe policy improvement (Swaminathan & Joachims, 2015; Dudík et al., 2011; Thomas et al., 2015; Laroche et al., 2019). Abstention is classical in selective prediction and learning to defer (Geifman & El-Yaniv, 2017; Madras et al., 2018; Mozannar & Sontag, 2020), and shift-aware conformal inference already handles covariate shift by importance weighting and bounds coverage loss without exchangeability (Tibshirani et al., 2019; Barber et al., 2023); our tools are standard and the two-environment argument behind Theorem 2 is textbook, what is new is the evidence hierarchy this assembles for agent memory, and the finding that its two levels are not equally attainable (Appendix A).

2 Memory Use as a Per-Pair Intervention

A request $x \sim \mathcal{D}$ induces a finite-horizon POMDP (equivalently its belief-state MDP). Injecting a retrieved memory m at episode start replaces the memory-free representation $\Phi_0(x)$ by $\Phi_1(x, m)$; it changes neither dynamics nor the tool set, only the inputs to a fixed policy π_θ. Most mechanisms agents use to exploit experience (episodic memory, warm-start, reflection, generative exploration) act through such input/parameter interventions. Existing reuse mechanisms typically do not attach coverage-dependent guarantees to the full-trajectory effect of injecting a specific memory, the deployment gap we address.

Definition 1 (Intervention outcome)**.** Under intervention $a \in \{0, 1\}$ the agent runs the whole episode and returns normalized utility $U_a \in [0, 1]$; ξ collects the initial state and model, decoding, and environment randomness. The baseline arm does not see m, so we write $U_0(x; \xi)$. Define $\Delta(x, m; \xi) = U_1(x, m; \xi) - U_0(x; \xi) \in [-1, 1]$ and the transfer gain $\tau(x, m) = \mathbb{E}_\xi[\Delta]$, relative to the fixed executor π_θ. The two arms run on *independently reset replicas* of the same initial state, so Δ is a paired potential-outcome contrast, not an inferred counterfactual. $\tau > 0$ helps; $\tau < 0$ is negative transfer; $\tau \approx 0$ adds no incremental task utility.

In words, $\tau(x, m)$ is not the quality of m in isolation. It asks whether giving *this* memory to *this* fixed agent on *this* request improves the full episode relative to giving it no memory at all.

We cannot estimate a separate effect for every request–memory pair, because almost no deployment pair has been practiced exactly. Generalization across pairs therefore runs through a structural description of the pair.

Definition 2 (Structural pair representation)**.** The gate input is $Z = h(x, m, \mathcal{T}) \in \mathcal{Z}$, not a request-only code, where $\mathcal{T}$ is the *static* tool schemas. Z encodes tool families, argument constraints, read/write structure, schema compatibility, and memory provenance, where provenance is the lexical test of whether m names the object x names, not an experimenter label; the complete encoder and a feature-by-feature observability table are in Appendix C. The same x with a supportive versus

corrupted m must yield different Z. We use exactly two encoders and label every claim with which one it uses: the *static* h, which consumes specifications only and carries TRACE-AVG and the base certified gate, and a *probe-augmented* h^+, used *only* by the TRACE-CERT+probe variant, which additionally reads a binned request-time estimate of the no-memory return from $R_0{=}4$ scored rollouts and therefore presupposes a resettable environment and an outcome scorer (Appendix C). The deployment pair law is the single-candidate proposal law: with $X \sim \mathcal{D}$ and index $J \sim \nu(\cdot \mid X)$ over the retrieved set, $(X, M_J) \sim P$; Q is the paired-practice law. Pair-level population statements use P; the deployed top-k pipeline is analyzed at two levels (candidate-set average value, §4.2; joint per-request coverage, §4.3).

Pairs sharing a value of Z are treated as statistically related evidence, which is why the representation is load-bearing rather than feature engineering: if a supportive and a corrupted memory collapse to the same Z, no downstream certificate can separate them.

Paired practice observations are $(X_i, M_i, \widetilde{\Delta}_i)$ with $(X_i, M_i) \sim Q$ i.i.d. and $\widetilde{\Delta}_i \in [-1, 1]$ the measured return difference. Matched seeds preserve the estimand and reduce variance only when the two arms are positively correlated; they do not cancel confounding. We write $\mu_Q(z), \mu_P(z)$ for the fiber-average effects among practice and deployment pairs at structure z, and audit the practice–deployment gap ϵ_{rep} and evaluator bias ϵ_{eval} as explicit terms rather than assuming them zero (Appendix E).

Reading key. x is the request and m a retrieved past experience; $\tau(x, m)$ is how much that memory changes the return; $z = h(x, m, \mathcal{T})$ is the pair's structural description; Q and P are the practice and deployment laws of structures; and $\widehat{\tau}(z)$, $s(z)$ are the estimated transfer and its evidence radius. Two questions follow throughout: what is the transfer effect, and is there enough coverage to act on it?

3 A Controlled Experience-Transfer Benchmark

To show relevance does not determine transfer we build a benchmark holding count and stated success fixed while making surface similarity work *against* a relevance rule. Each of 16 tool domains has a read-only and a destructive tool over an identified object (8 practice, 8 held-out deployment); requests come at three ambiguity levels crossed with phrasing variants. Every task runs under *none* plus eight memory types with $R{=}3$ repetitions, two of which carry the mechanism: lexical distractor (same object, wrong action) and structural match (different object, same workflow). Candidates are matched on count and on stated source-task success rate, but *not* on surface similarity: matching that would defeat the purpose. Mean token-overlap (Jaccard) similarity to the request is deliberately adversarial, and identical for the harmful lexical distractor and the helpful supportive memory (0.061 each), while the helpful structural match sits *lower* (0.037). Any similarity threshold that accepts the supportive memory therefore accepts the distractor as well, which is the property we exploit rather than a confound we remove. Paired rollouts start from the same reset state and scoring is deterministic execution ($\epsilon_{\text{eval}}{\approx}0$, no LLM judge): a controlled contrast isolating structure, not a causal claim.

4 Coverage-Calibrated Transfer

Coverage creates three questions: how accurately practice estimates the deployment value of a rule where the two overlap, when there is enough local evidence to claim anything about one particular pair, and what to do when the deployment structure was never practiced at all. Theorem 1 answers the first, the certified gate (§4.3) the second, Theorem 2 the third.

The third answer is the binding one: where practice covers no deployment structure, no method identifies the transfer sign, so a mandatory binary gate is provably lossy under an oracle-regret objective, which is what motivates abstention. Wherever coverage *does* exist, what remains is to say exactly how strong a claim it supports, and we keep two answers apart. A candidate-set pessimistic selector guarantees non-negative average policy value on a pre-specified *practice-side* target, with deployment transfer controlled by a declared sensitivity term ($\epsilon^{C}_{\text{rep}}$ for the i.i.d. target, $\epsilon^{C,\text{grid}}_{\text{rep}}$ for

the fixed design) (§4.2); a conformal certificate gives *marginal*, not conditional, coverage of the pair-level effect (§4.3). Full proofs are in Appendices E–J.

What can be estimated from a single practice sample. Practice and deployment need not contain structural pairs in the same proportions, so practice observations must be reweighted. The importance weight $w(z) = \frac{dP_Z}{dQ_Z}(z)$ asks how much more often structure z matters at deployment than it appeared in practice: a structure that is 10% of practice but 30% of deployment receives roughly three times the weight. The deployment value of a gate g is $G_P(g) = \mathbb{E}_P[g(Z)\tau]$, estimated by $\widetilde{G}_N(g) = \frac{1}{N}\sum_i w(Z_i)g(Z_i)\widetilde{\Delta}_i$. Coverage enters through the *second-moment* coefficient $C_2 = \sup_{g\in\mathcal{G}} \mathbb{E}_Q[w(Z)^2 g(Z)]$, which replaces the worst-case $\|w\|_\infty$ in the leading variance term, so a few large weights inflate variance rather than dominating the bound (B still appears in the $1/N$ tail). There are then only four ways practice can mislead us about deployment: finite data, rare structures carrying large weights, a practice-to-deployment mismatch inside the representation, and evaluator error. Theorem 1 makes all four explicit.

Theorem 1 (Uniform target-gate bound)**.** *Let $t_\delta = \log(2|\mathcal{G}|/\delta)$ and B bound the absolute value of the weighted summand, so the centered summand is bounded by $2B$. Under overlap, representation-mismatch, and frozen-round assumptions (Appendix E), with probability $\geq 1-\delta$,*

$$\sup_{g\in\mathcal{G}} \left|\widetilde{G}_N(g) - G_P(g)\right| \leq \underbrace{\sqrt{\tfrac{2C_2 t_\delta}{N}}}_{\textit{noise}} + \underbrace{\tfrac{4Bt_\delta}{3N}}_{\textit{heavy-weight tail}} + \underbrace{\epsilon_{\mathrm{rep}}}_{\textit{shift}} + \underbrace{\epsilon_{\mathrm{eval}}}_{\textit{evaluator}}.$$

The estimator error scales as $n_{\mathrm{eff}}^{-1/2}$ with the exact effective sample size $n_{\mathrm{eff}}(g) = Np_g^2/\mathbb{E}_Q[w^2 g]$; $\epsilon_{\mathrm{rep}}, \epsilon_{\mathrm{eval}}$ are audited, not assumed zero.

Interpretation. The first two terms are statistical and shrink as effective data accumulates; the last two are biases that more sampling does not remove. Coverage reads most easily through $n_{\mathrm{eff}}(g) = Np_g^2/\mathbb{E}_Q[w^2 g]$, the practice that actually supports this gate once deployment mismatch is corrected for, so better coverage means larger n_{eff} and smaller error. Where a structure is uncovered the bound is vacuous, the regime §4.4 formalizes.

4.1 Transfer estimation under coverage

For a candidate at structure z with estimate $\widehat{\tau}(z)$ and a z-measurable uncertainty score $s(z) \geq 0$ (fixed with $\widehat{\tau}$ on folds disjoint from selection), the scale-λ gate is $g_\lambda(z) = \mathbf{1}\{\widehat{\tau}(z) - \lambda\, s(z) > 0\}$. The three quantities have distinct roles: $\widehat{\tau}$ says what we think the effect is, s says how much evidence supports that estimate, and λ says how conservatively we act on that evidence. An estimate of $+0.3$ with radius 0.05 is actionable; the same $+0.3$ with radius 0.5 is not. Two instantiations matter: the analytic radius $s_{\mathrm{analytic}}(z) = r(z,\delta) + \beta_\gamma(z)$ (smoothness plus fiber envelope) and the deployed conformal radius $s_{\mathrm{conformal}}(z) = \widehat{\beta}_\gamma(z) + \rho_R$; experiments use the conformal score. Whether $\lambda s(z)$ is a *valid confidence radius* (supporting per-pair claims) is separate from selecting λ for *average* value: we keep two grids, $\Lambda_{\mathrm{AVG}} \subset (0,\infty)$ for the average-value selector and $\Lambda_{\mathrm{CERT}} \subset [1,\infty)$ for certified deployment ($\lambda < 1$ shrinks the interval below a confidence set, destroying per-pair coverage).

4.2 Pessimistic average-value selection

The deployed object is a policy over candidate sets $C = (X, M_1, \ldots, M_k)$: π_λ shortlists, compares lower confidence bounds, and returns an action in $\{0,\ldots,k\}$ (no-memory or one candidate). This regime does not try to certify every accepted memory. Among a finite set of candidate policies it selects the one whose estimated average gain is largest *after* subtracting a statistical penalty,

$$\widehat{\pi} = \arg\max_{\pi\in\Pi} \left\{\widehat{J}_N(\pi) - R_N(\pi)\right\},$$

so it takes the best worst-case value rather than the highest point estimate. The no-memory policy has known value zero, which makes it a safety anchor: when in doubt, do not use a memory.

Proposition 1 (Pessimistic candidate-set selection)**.** *Let $C_1, \ldots, C_N \sim Q_C$ i.i.d. be practice candidate-set contexts (frozen retriever), and $J(\pi) = \mathbb{E}_{Q_C}[U_{\pi(C)} - U_0] \in [-1,1]$ the policy's*

paired incremental value, estimated by $\widehat{J}_N$ from measured paired rollouts. For the finite class $\Pi = \{\pi_\lambda : \lambda \in \Lambda_{\mathrm{AVG}}\} \cup \{\pi_0\}$ ($\pi_0 \equiv$ no-memory, $J(\pi_0)$=0), with all maps frozen on disjoint folds, $R_N = \sqrt{2\log(2|\Pi|/\delta)/N}$, and $\widehat{\pi} = \arg\max_\pi \{\widehat{J}_N(\pi) - R_N \mathbf{1}\{\pi \neq \pi_0\}\}$: with probability $\geq 1-\delta$, $\widetilde{J}_{Q_C}(\widehat{\pi}) \geq 0$ and $\widetilde{J}_{Q_C}(\pi) - \widetilde{J}_{Q_C}(\widehat{\pi}) \leq 2R_N$ for all π, where $\widetilde{J}$ is the measured *value. Two audited terms carry the conclusion to the true target:*

$$J_{Q_C}(\widehat{\pi}) \;\geq\; -\epsilon^C_{\mathrm{eval}}, \qquad J_{P_C}(\widehat{\pi}) \;\geq\; -\epsilon^C_{\mathrm{eval}} - \epsilon^C_{\mathrm{rep}},$$

with the oracle inequality inflating by $2\epsilon^C_{\mathrm{eval}} + 2\epsilon^C_{\mathrm{rep}}$.

Here $\epsilon^C_{\mathrm{eval}}$ is the candidate-set evaluator bias, the gap between the measured and true paired value, and $\epsilon^C_{\mathrm{rep}}$ is the practice-to-deployment transport error; both are defined formally in Appendix I.3. Under an exact evaluator $\epsilon^C_{\mathrm{eval}}$=0, which holds for our controlled benchmark (deterministic execution scored against ground truth), the guarantee recovers $J_{Q_C}(\widehat{\pi}) \geq 0$.

Corollary 1 (Variance-adaptive confirmatory selection)**.** *On the pre-fixed confirmatory grid, Proposition 1 remains valid with the common Hoeffding penalty replaced by an independent, non-i.i.d. empirical-Bernstein radius $R^-_N(\pi)$ (Maurer & Pontil, 2009, Thm. 11), which is tighter whenever observed policy returns have low variance. The exact radius, the $[-1,1] \to [0,1]$ transformation, the two-sided oracle inequality, and the fixed-grid justification are in Appendix I.4. On our confirmatory fold this selector chooses $\lambda^\star$=0.25.*

$\epsilon^C_{\mathrm{rep}}$ is a declared *sensitivity parameter*, not an estimate: cell-mean agreement between practice and deployment is supporting evidence, not a valid upper bound on it. Per-pair claims belong exclusively to §4.3.

4.3 Pairwise transfer certification

For a fixed structure z, center split-conformal residuals of repeated-rollout pair means at the deployed estimator $\widehat{\tau}(z)$; the $\lceil (n_z{+}1)(1-\gamma) \rceil$-th order statistic gives an envelope $\widehat{\beta}_\gamma(z)$ ($+\infty$ when the index exceeds n_z: too little data to certify, so the system defers), and adding the Hoeffding rollout radius $\rho_R = \sqrt{2\log(2/\gamma')/R}$ (range 2) certifies $|\tau(X,M) - \widehat{\tau}(z)| \leq \widehat{\beta}_\gamma(z) + \rho_R$ with probability $\geq 1-\gamma-\gamma'$ (Appendix I). This is a *marginal* guarantee over a random deployment pair, and rests on a residual-transport assumption (within-cell residual law carries from Q to P) that we audit, not prove.

Proposition 2 (Joint top-k certificate, conformal form)**.** *Fix $\lambda \geq 1$ and total error α. With shortlist and $\widehat{\tau}$ frozen, allocate $\gamma_c = \gamma_r = \alpha/(2k)$ and set $r_j = \widehat{\beta}_{\alpha/(2k)}(z_j) + \sqrt{2\log(4k/\alpha)/R}$. Then simultaneously for all j, $|\tau(x,m_j) - \widehat{\tau}(z_j)| \leq \lambda r_j$ with probability $\geq 1-\alpha$, and the use/fallback/defer branches carry their stated meanings.*

The radius has two parts. The envelope $\widehat{\beta}$ accounts for how much individual pair effects vary around the structural prediction; the $R^{-1/2}$ term accounts for noise from estimating one pair effect with finitely many repeated rollouts. A certificate radius is therefore between-pair uncertainty plus within-pair rollout uncertainty.

Remark 1 (Two regimes, two names). The certified gate ($\lambda \geq 1$, Proposition 2) is the only regime that may be called *certified*. The average-value selector (any $\lambda \in \Lambda_{\mathrm{AVG}}$, Proposition 1) carries the candidate-set average guarantee but *no* per-pair claim. The regimes trade coverage of the upside against the strength of the per-decision claim.

4.4 Coverage-hole impossibility

Consider a deployment region that never appears in practice (Figure 4, Appendix I.10). Two worlds are then observationally identical before deployment: in one, memory helps by $+\Delta$ on that region; in the other it hurts by $-\Delta$. Because the pre-deployment data have the same law in both, no algorithm can tell which world it is in.

Theorem 2 (Coverage-hole minimax impossibility)**.** *Measure quality by the memory-decision regret against the oracle gate, $\mathcal{R}_M(g) = \mathbb{E}_{P_Z}[\max\{0, \mu_{P,M}(Z)\} - g(Z)\mu_{P,M}(Z)]$. Fix a gap $\Delta \in (0,1]$*

and a mandatory binary gate. If a region $A \subseteq \mathcal{Z}$ has $P_Z(A) \geq \alpha$ but $Q_Z(A) = 0$, and two environments $M_\pm$ agree outside A with $\mu_{P,M_\pm}(z) = \pm\Delta$ on A while all pre-deployment information has identical law, then every data-dependent gate obeys $\inf_{\widehat{g}} \sup_{M \in \{M_+, M_-\}} \mathbb{E}[\mathcal{R}_M(\widehat{g})] \geq \alpha\Delta/2$.

Here α lower-bounds how often the unseen region occurs at deployment and Δ is how costly getting its sign wrong can be. Where practice gives no evidence about a deployment structure, no method identifies the transfer sign, and abstention is not merely permitted but *optimal* once its price is below what guessing costs:

Corollary 2 (When deferral is minimax-optimal)**.** *Add a* defer *action routing the pair to a channel that resolves it at cost $c_d \geq 0$, and write $p_A = P_Z(A) \geq \alpha$. On A the minimax regret of any mandatory binary rule is $p_A\Delta/2$, while deferring costs $p_A c_d$. Hence deferral strictly dominates every mandatory binary gate when $c_d < \Delta/2$, ties the binary optimum at $c_d = \Delta/2$, and is dominated when $c_d > \Delta/2$. The threshold is free of p_A, and outside A the two action sets coincide. The resolution premise is load-bearing and our deployed rule does not satisfy it; see below.*

Operational qualification. The resolution premise is load-bearing: c_d prices a channel that *resolves* the current request. If deferral instead falls back to the no-memory arm, as in our own deployment (§6), it is not a third action but the discard branch of a binary gate: correct on M_-, forfeiting the whole gain on M_+, so its worst case is $p_A\Delta$ with any acquisition cost added rather than substituted. Such a rule is therefore worse than the randomized binary optimum $p_A\Delta/2$, and deferral there should be read as an *acquisition signal* whose value accrues to later requests, a sequential claim this single-request statement does not make.

This compares two prices: deferring costs c_d per unresolved pair, while guessing an unidentified sign costs at least $\Delta/2$ in the worst case. So the architecture is motivated by a lower bound rather than by intuition. Active repair (§5) is the *sequential analogue* of this logic: it acquires evidence for future requests sharing the structure rather than guessing through the hole, which is a different action from the resolving deferral priced at c_d above. Deferral is structural, not a heuristic.

The deployment pipeline. Deployment runs a fold-structured recipe on disjoint folds (full version in Appendix L): **A0** fixes h, the metric and the cell partition before any calibration, so exchangeability is not broken by tuning cells to outcomes; **A1** fits $\widehat{\tau}(z)$ and the shortlist map; **A2** freezes both and forms $\widehat{\beta}_\gamma(z)$ and the per-pair radius; **fold C** selects λ pessimistically (Corollary 1). At deployment we form $[\widehat{\tau}(z_j) \pm \lambda r_j]$ per candidate and use the best LCB if it exceeds 0, fall back if every UCB is below 0, and otherwise *defer*, with $\lambda \in \Lambda_{\mathrm{CERT}}$ under the $\alpha/(2k)$ allocation of Proposition 2 for the certified gate and the fold-C $\widehat{\lambda}$ for the average-value selector. Three rules share this interface and differ only in what they may claim; Table 3 gives each one's guarantee, assumption and evidence cost.

5 Active Coverage Repair

Deferral is an acquisition signal, not a terminal state. When a cluster of pairs is frequent (high P_Z), sign-unresolved (the interval straddles zero), and wide, we spend a paired-practice budget there, refit the estimator and envelope on fresh folds, and re-decide. We optimize a monotone-submodular *frozen-round surrogate* for the deployment-weighted statistical component of width reduction, so greedy acquisition attains the classical $(1 - 1/e)$ guarantee (Nemhauser et al., 1978) against that surrogate's budget-constrained optimum. Two limits belong here rather than in the appendix. The guarantee covers the surrogate, not the full closed loop with its refitting and bias terms. And acquisition tightens intervals without licensing a per-pair claim: every post-repair action we measure is taken by the average-value selector at $\lambda < 1$, carrying Proposition 1, not Proposition 2. Budgets, width curves and the random-allocation control are in Appendix J.

6 Experiments

The experiments are not a leaderboard: each checks a specific prediction of §4, and the tightest coupling is the one that is unflattering when it holds, since the certified regime certifies nothing at the initial evidence level ($0/32$). Appendix D maps every claim to its experiment.

Memory type	$\widehat{\tau}$	95% CI	Memory type	$\widehat{\tau}$	95% CI
supportive	$+0.167$	$[+0.10, +0.24]$	conflicting	$+0.139$	$[+0.07, +0.21]$
structural match	$+0.167$	$[+0.10, +0.24]$	irrelevant	-0.014	$[-0.12, +0.09]$
stale	$+0.167$	$[+0.10, +0.24]$	**lexical distractor**	$\mathbf{-0.247}$	$[-0.34, -0.17]$
partial	$+0.163$	$[+0.09, +0.25]$	**corrupted**	$\mathbf{-0.420}$	$[-0.50, -0.33]$

Table 1: **Structural incompatibility, not similarity, predicts harmful transfer.** The harmful lexical distractor is *more* surface-similar than its benign structural-match twin (0.061 versus 0.037 mean token overlap). Paired transfer on 768 practice pair units, task-clustered bootstrap; domain-clustered and leave-one-domain-out versions in Appendix V.

External validity is not a single claim here; we separate controlled mechanism isolation, executor replication, and benchmark transport. Organic benchmarks jointly vary memory relevance, memory quality, executor competence, and task difficulty, which is what makes the independent role of structural compatibility hard to isolate, so the controlled benchmark is deliberately diagnostic: it holds count and source-task success fixed while making the harmful candidate the *more* surface-similar one, so a similarity rule cannot succeed by construction rather than by accident. It therefore does *not* estimate the prevalence of harmful memory in the wild. Cross-family runs then ask whether the sign pattern is executor-specific, and BFCL/AppWorld test transport to public environments, one a ceiling/abstention regime and one a boundary where the twin trap does not reproduce. Evaluation runs in three stages: a ground-truth simulator verifying every finite-sample prediction (Appendix K); the controlled benchmark on live APIs, paired on reset replicas, with Claude Haiku 4.5 as primary executor, the full benchmark replicated on Qwen3-32B and GPT-OSS-120B and the blind slice on Claude Sonnet 4.5; and mechanism-replication and scope tests. Retrieval returns $k{=}8$ candidates, of which we shortlist the top $K{=}2$ by train-fold $\widehat{\tau}$ under the $\alpha/(2K)$ allocation of Proposition 2 (Appendix L).

Structure, not similarity, predicts transfer (Table 1). Only the two structurally incompatible types are reliably harmful. A structural match shares the estimate of a supportive memory, but both saturate at 1.000, so their per-task contrast is identically zero and we claim no effect equivalence. Against a 11.3% harmful base rate, similarity scores 0.086 AUPRC and an LLM judge 0.112, both uninformative, against 0.244 for a structure-aware supervised score.

TRACE-AVG matches the oracle; TRACE-CERT abstains (Table 2). The certified gate ($\lambda{=}1$) defers everywhere at the initial evidence level: one calibration round leaves within-cell heterogeneity no per-pair sign is certifiable against. An exploratory refinement, adding a decision-time probe band to h^{+}, produced $13/32$ nominal uses *on the data that motivated it*. We then tested it properly: with h^{+}, its binning and the split frozen in advance, an independent $46{,}080$-episode fold on fresh, previously unused tasks produced $0/32$ **uses and** $32/32$ **deferrals** at utility 0.496, exactly never-memory. No split produced a single use, and no deployment-relevant cell retained the 39 residuals $\alpha{=}0.1$ demands, so the opening does not replicate. A post-hoc $K{=}1$ diagnostic shows multiplicity was only part of it: it ceases to bind, yet every lower bound still sits just below zero, leaving within-cell heterogeneity as the constraint (Appendix T). The protocol that yields an actionable average-value guarantee yields no per-pair certificate.

The average-value claim opens on practice, then on a held-out-domain grid. At $N{=}96$ neither theorem radius admits a non-trivial selection, so only a bootstrap picks $\lambda^{\star}{=}0.25$. A pilot/confirmatory protocol fixes that: demote all prior practice to a pilot that freezes the estimators and the fold size (Corollary 1) before collection, then evaluate a fresh disjoint 384-task fold *once*. The empirical-Bernstein value is positive ($\widehat{J}_C{=}0.163$, margin $+0.039$) and the *formal* selector picks $\lambda^{\star}{=}0.25$ (Table 2). That statement is practice-side, so the deployment bound it implies is $0.039 - \epsilon_{\mathrm{rep}}^{C,\mathrm{grid}}$, a *declared sensitivity parameter* we do not estimate (Appendix N). We therefore measured the held-out domains rather than assuming transport: holding the encoder, cells, envelope and λ fixed and committing every per-task decision to a manifest beforehand, we evaluated that *single* policy on a fresh fixed-design grid of 384 tasks over the eight held-out deployment domains. One policy means $|\Pi|{=}1$ and no union bound, so $\widehat{J}_D{=}+0.120$ gives $\mathrm{LCB}_{0.95}[J_D(\pi)]{=}+0.032$ for that grid with no transport term, at $128/384$ uses and 0 harmful (exact upper 0.023).

Panel A: operational comparison on 96 held-out-domain tasks

Rule	Utility ↑	Harmful uses ↓	Uses	Extra online evals / request	Statistical claim
never memory	0.826	—	$0/96$	0	—
similarity	0.979	$2/96$	$96/96$	0	none
LLM judge	0.885	$7/96$	$96/96$	1 judge call	none
CMI (Srivastava, 2026)	1.000	$0/19$	$19/96$	51 probes + 51 scores	none
point estimate only	1.000	$0/96$	$96/96$	0	none
TRACE-AVG	1.000	$0/32$	$32/96$	0	**average-value**
TRACE-CERT	0.826	—	$0/96$	0	pair-level; no openings
oracle	1.000	—	—	—	—

Panel B: evidence status

Claim	Evaluation protocol	Result	Status
average-value	fresh 384-task practice fold	practice-side LCB $+0.039$	**supported**
average-value	frozen policy, 384-task *held-out-domain* grid	grid LCB $+0.032$, no transport term	**supported**
pair-level + probe	representation selected post-hoc	$13/32$	exploratory
pair-level + probe	frozen 46,080-episode validation	$0/32$	**not replicated**

Table 2: **Main empirical result.** TRACE-AVG matches oracle utility at zero observed harm, and its average-value claim opens twice: on a fresh practice fold and, with the policy frozen beforehand, on a fixed-design grid over the eight held-out domains, with no transport term (Appendix O). Pair-level certification does not open, and its exploratory opening fails frozen validation. *Point estimate only* drops the radius and ties TRACE-AVG here, so the radius buys abstention, not utility; under candidate-set removal its harm rises to $12/32$ while TRACE-AVG holds 0 (Appendix U). Harmful uses count only uses taken; CMI gets an oracle scorer it would lack in deployment.

As with J_C, J_D is a grid mean, and the 0.043 gap is an observed selected-policy gap, not an estimate of the supremum transport error (Appendix O).

Mechanism replication, scaling, and closest-method comparison (Table 11). Appendix P gives the cross-family grid (Table 8): the sign pattern replicates on Qwen3-32B, GPT-OSS-120B and the Sonnet 4.5 blind slice, aggregate conformal coverage meets nominal on all three, and one adversarial slice reveals a transport failure. Within Qwen3 the distractor penalty is monotone in scale ($-0.35 \to -0.49 \to -0.67$, 8B→32B): larger models follow wrong memory more faithfully, so scale amplifies rather than mitigates it. On BFCL V4 multiple the pattern replicates (distractor -0.087, supportive $+0.032$) in an explicit-request ceiling regime where our selector and a supervised gate both correctly abstain. Table 2 places CMI beside us: equal utility and harm, bought with 51 probes and 51 scorer calls per request where we issue none. Adapted RSCB-MC and ToolChain-CRC proxies track always-use and stay diagnostic (Appendix Q).

Organic retrieval yields no reliable negative-transfer regime. Removing our authorship of the memory text removes the reliable harm. Three earlier organic probes found the mismatched candidate inert or helpful. A preregistered Step-0 AppWorld pilot retrieved verbatim benchmark-authored workflows into untouched, non-ceiling test tasks and failed its prespecified harm gate. An independent five-repetition block reverses both of its two negative outcomes, the larger from -0.700 to $+0.220$ (Appendix X). A second manifest-frozen pilot compared verbatim same-app workflows classified by a train/dev-frozen ontology as aligned or opposite-direction. Both beat no memory ($+0.157$ and $+0.119$), and their difference was small and unresolved (-0.038, $[-0.107, +0.021]$; Appendix Y). The harmful class we can measure *reliably* is thus the one we construct.

7 Conclusion

What a memory is worth is its incremental transfer value, not its relevance, and coverage sets what may be claimed. The average-value guarantee opens on fresh practice data and, with the policy frozen beforehand, on a fresh grid over the held-out deployment domains; independently validated pair-level certification was not, at the evidence level we could afford. Four limits bound this: an episode-level decision unit rather than a sequential policy; a bound for the fixed-design grid we specified, not for arbitrary shift; eight domains; and reliable harm only in memories we authored, with none of five organic settings producing replicable harm (Appendix S).

Ethics Statement

This work studies when reusing an agent's own past experience degrades its behaviour, and the mechanism it isolates is dual use: the same construction that lets us measure harmful transfer also describes how a memory store could be poisoned to redirect an agent toward a destructive tool call (Chen et al., 2024). We judge publication to be net positive because the contribution is a *defensive* decision rule that abstains when evidence is thin, and because the failure mode is already documented in the literature we build on. All harmful memories are authored by us and injected inside sandboxed benchmarks; the destructive tools are simulated, no live system or third-party account is acted upon, and no human subjects, personal data, or private data are involved. We report the one audited case where our own transport assumption fails rather than omitting it, since a certificate presented as stronger than it is would be the most harmful outcome of this work.

AI Use Statement

Generative AI tools were used to assist with brainstorming, code development, manuscript drafting and editing, literature search, and checks of mathematical exposition. All research decisions, theorem statements, proofs, experiments, citations, and reported numerical results were independently reviewed and verified by the authors against source materials and experiment logs. The authors take full responsibility for the manuscript and associated artifacts.

Reproducibility Statement

All utility statistics are deterministic functions of append-only experiment logs keyed by (task, memory kind, repetition, model). Track AA's process audit is distributed as a credential-free per-run derived cache because the raw API logs contain synthetic passwords and access tokens. Every reported statistic is therefore recomputable without additional model calls, and two runs of the analysis return identical numbers. The material needed to reproduce the work is in the appendix rather than restated here: benchmark construction, the sixteen domains and eight memory types with their templates (Appendix L); assumptions, estimator definitions, and complete proofs of every theorem, proposition, and corollary (Appendices E–I); fold assignments and their disjointness, with the pilot (192 practice tasks), confirmatory fold C (384 fresh practice tasks), fold D (a 384-task fixed-design grid over the eight held-out deployment domains, whose frozen per-task decisions are committed to a manifest before collection; Appendix O), and the untouched 96 held-out-domain deployment tasks carrying no role overlap (Appendix N); calibration protocol, rollout counts, and decision rules (Appendices M, N); baseline implementations and the adaptations we made to them (Appendix Q); and a claim-to-experiment roadmap mapping each reported result to the script and log that produce it (Appendix D). The simulator verification of every finite-sample prediction (Appendix K) requires no API access at all. The encoder and its cell partition are given as pseudocode with a feature-by-feature observability table (Appendix C), and the sixteen domains, the eight memory types and the memory-note schema are described in Appendix L. The verbatim system prompt, the full tool-schema JSON, and the memory-note templates are *not* reproduced in the paper for length reasons; they are deterministic outputs of the released benchmark generator, and we provide them together with the append-only logs as anonymized supplementary material rather than asking a reader to reconstruct them from the appendix.

References

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2208.02814.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1645–1661, 2025. doi: 10.1214/24-AOAS1998. arXiv:2110.01052.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. doi: 10.1214/23-AOS2276.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 1042–1051, 2019.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 130185–130213, 2024. arXiv:2407.12784.

Yumin Choi, Sangwoo Park, Minki Kang, Jinheon Baek, and Sung Ju Hwang. PREPING: Building agent memory without tasks. *arXiv preprint arXiv:2605.13880*, 2026.

Mahavir Dabas, Jihyun Jeong, Ming Jin, and Ruoxi Jia. Memory-induced tool-drift in LLM agents. *arXiv preprint arXiv:2605.24941*, 2026. ICML 2026 AIWILD workshop.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 1097–1104, 2011.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4878–4887, 2017.

Xinyu Guan, Qianyang Zhao, and Yuming Deng. Decision-aware memory cards: Counterfactual-inspired context selection and compression for tool-using LLM agents. *arXiv preprint arXiv:2606.08151*, 2026.

Qisheng Hu, Quanyu Long, and Wenya Wang. When continual learning moves to memory: A study of experience reuse in LLM agents. *arXiv preprint arXiv:2604.27003*, 2026.

Mehmet Iscan. Learning when to remember: Risk-sensitive contextual bandits for abstention-aware memory retrieval in LLM-based coding agents. *arXiv preprint arXiv:2604.27283*, 2026.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 3652–3661, 2019.

Lihua Lei and Emmanuel J. Candès. Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society: Series B*, 83(5):911–938, 2021.

Ziyang Luo, Zhiqi Shen, Wenzhuo Yang, Zirui Zhao, Prathyusha Jwalapuram, Amrita Saha, Doyen Sahoo, Silvio Savarese, Caiming Xiong, and Junnan Li. MCP-Universe: Benchmarking large language models with real-world model context protocol servers. *arXiv preprint arXiv:2508.14704*, 2025.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6147–6157, 2018.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample-variance penalization. In *Proceedings of the 22nd Conference on Learning Theory (COLT)*, 2009. arXiv:0907.3740.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 7076–7087, 2020.

Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14(1):265–294, 1978.

Jeffery Opoku and David Banahene. ToolChain-CRC: Conformal risk control for agentic AI under retrieval and tool-use drift. *arXiv preprint arXiv:2606.18467*, 2026.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pp. 1–22, 2023.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 48371–48392, 2025.

Shuai Shao, Qihan Ren, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, Dongrui Liu, and Jing Shao. Your agent may misevolve: Emergent risks in self-evolving LLM agents. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.26354.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.03768.

Saksham Sahai Srivastava. Causal intervention-based memory selection for long-horizon LLM agents. *arXiv preprint arXiv:2605.17641*, 2026.

Iiris Sundin, Peter Schulam, Eero Siivola, Aki Vehtari, Suchi Saria, and Samuel Kaski. Active learning for decision-making from imbalanced observational data. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6046–6055, 2019.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 814–823, 2015.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 2380–2388, 2015.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2526–2536, 2019.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024. arXiv:2407.18901.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 63897–63911, 2025. arXiv:2409.07429.

Hechuan Wen, Tong Chen, Mingming Gong, Li Kheng Chai, Shazia Sadiq, and Hongzhi Yin. Enhancing treatment effect estimation via active learning: A counterfactual covering perspective. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pp. 66437–66466, 2025.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026. arXiv:2505.16067.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.12045.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. MemGen: Weaving generative latent memory for self-evolving agents. In *International Conference on Learning Representations (ICLR)*, 2026a. arXiv:2509.24704.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, Dat Huynh, Hengduo Li, Zi Yang, Sara Cao, Lawrence Jang, Shuyan Zhou, Jiacheng Zhu, Huan Sun, Jason Weston, Yu Su, and Yifan Wu. Agent learning via early experience. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026b. arXiv:2510.08558.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024. arXiv:2308.10144.

WHERE TO FIND EACH PROOF

Every result stated in the main text is proved below. The two impossibility- and selection-critical proofs are §I.10 and §I.4.

Main-text result	What it gives	Proof	Supporting
Thm 1 (uniform bound)	estimation error, n_{eff}	§F.1	§E, G, H
Prop 1 (the average-value selector)	average value ≥ 0	§I.3	§I intro
Cor 1 (EB selector)	variance-aware, fixed design	§I.4	§N (protocol)
Prop 2 (the certified gate)	per-pair marginal coverage	§I.8	§M (round)
Thm 2 (coverage hole)	regret $\geq \alpha\Delta/2$	§I.10	§J (acquisition)

Assumptions are stated once in §E and referenced by name thereafter; the two audited bias terms $\epsilon_{\text{rep}}, \epsilon_{\text{eval}}$ are defined there and carried explicitly through every bound. Empirical protocols are separate from proofs: §K (simulator checks), §L (full protocol), §M, §N, §P, §Q, §R, §Z.

A RELATED WORK

Intervention-based memory utility. Closest to us, concurrent work scores memory by *effect* rather than relevance: CMI uses request-time no-memory/with-memory/perturbed answer probes (Srivastava, 2026), and CICL ranks context by action shift, outcome uplift, necessity and negative-transfer risk (Guan et al., 2026). We share the shift but differ in three ways: the target is the *paired full-trajectory return* of a fixed executor rather than answer-level or judge-predicted uplift; it is estimated from *pre-deployment* paired self-practice through a structural representation rather than request-time gold-scored probes; and coverage-dependent guarantees are attached to it. **Selective and risk-sensitive control.** MemGen learns a memory trigger (Zhang et al., 2026a); RSCB-MC chooses among memory and non-memory actions from structural, uncertainty, feedback and cost features (Iscan, 2026), so it too uses structure, and our distinction is the *statistical target* rather than a learned reward or risk.

Abstention and learning to defer. A reject option is classical in selective prediction (Geifman & El-Yaniv, 2017), and learning-to-defer makes the abstention decision itself learnable (Madras et al., 2018; Mozannar & Sontag, 2020). Those methods route the current input to an expert whose error rate prices the deferral; ours routes a *structural region* to targeted practice, so the price is the effect size the region hides (Corollary 2) and resolution accrues to later requests.

Safe selection and calibration. More broadly, conformal risk control and learn-then-test calibrate decision parameters to control an expected loss (Angelopoulos et al., 2024; 2025); our target is instead the memory-versus-no-memory paired contrast. ToolChain-CRC calibrates *absolute* trajectory risk (Opoku & Banahene, 2026); we calibrate the memory-versus-no-memory *contrast*. The average-value selector specializes high-confidence safe policy improvement (Thomas et al., 2015; Laroche et al., 2019) with the known-zero no-memory policy as anchor, and the certificate is related to conformal treatment-effect inference (Lei & Candès, 2021) but must handle repeated agent rollouts, structural coarsening and practice-to-deployment residual transport. Active treatment-effect learning (Sundin et al., 2019; Wen et al., 2025) likewise acquires observations where the effect estimate is least informative; our repair specializes this to paired agent practice triggered by unresolved certificates.

Coverage shift and conformal validity. Our practice-to-deployment analysis is related to importance-weighted conformal calibration under covariate shift (Tibshirani et al., 2019) and to robustness bounds under non-exchangeability (Barber et al., 2023). We differ in calibrating a *paired causal contrast* over structural pairs rather than an emitted output, and in treating the residual transport gap as an audited sensitivity term rather than a corrected weight.

Practice, failure evidence, foundations. PREPING builds pre-deployment procedural memory (Choi et al., 2026), on top of which we gate and repair. Experience-following, tool-drift and self-evolution risk (Xiong et al., 2026; Dabas et al., 2026; Shao et al., 2026) document closely related memory-induced failure modes. Foundations: off-policy coverage (Swaminathan & Joachims, 2015; Dudík et al., 2011; Munos & Szepesvári, 2008), selective prediction and conformal calibration (Geif-

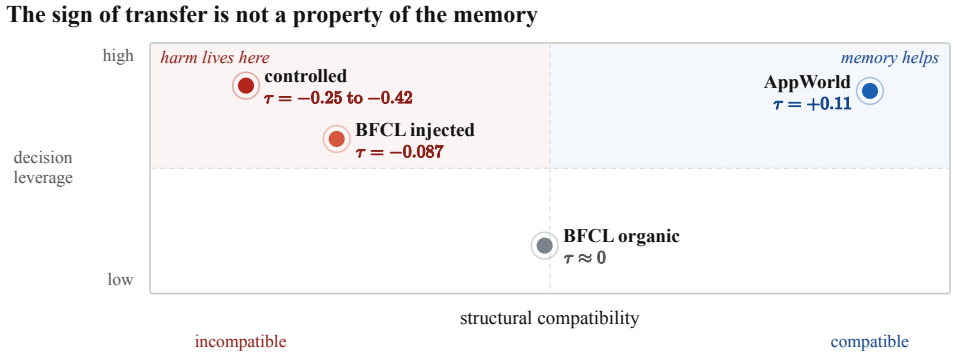


Figure 2: Our four settings on the two axes that organize the transfer regimes in our controlled study: how much leverage the injected memory has over the trajectory, and whether its structure is compatible with the request. In our controlled harmful regime negative transfer concentrates in this quadrant, and a retrieval label does not tell you which quadrant you are in. The axes are a description of where we observed harm, not a proven characterization: on AppWorld a structurally mismatched workflow was helpful on average, which is the boundary case discussed in §R.

man & El-Yaniv, 2017; Angelopoulos & Bates, 2023), and agent benchmarks (Patil et al., 2025; Trivedi et al., 2024).

B THE FOUR REGIMES

What each rule may claim. The three deployed rules share one interface, one estimator and one encoder. They differ in the scale λ, in the assumption that scale requires, and therefore in the strength of the statement they support and the evidence it costs (Table 3).

	TRACE-AVG	**TRACE-CERT**	**CERT + probe**
target	candidate-set mean J	this pair's τ	this pair's τ
guarantee	average value (Prop. 1)	marginal per-pair (Prop. 2)	nominal only: h^+ chosen post hoc
λ	< 1, fold-C selected	≥ 1, $\alpha/(2k)$	≥ 1, $\alpha/(2k)$
assumption	fixed-design target; fold C needs ϵ_{rep}, fold D (direct) none	residual transport within the cell	as CERT, plus resettable env. and request-time scorer
evidence	one fresh fold	$\geq$39 residuals *per cell*	same, plus probes
outcome here	**opens**: folds C, D	**never opens**: $0/32$	$13/32$, not replicated

Table 3: Only the average-value rule opened at the evidence level we could afford. Its transport term is needed to carry the fold-C bound to deployment and vanishes only for the direct, frozen-policy fold-D grid evaluation. The probe variant's representation was selected post hoc on the same task set, so its $13/32$ opening is nominal rather than certified, and it did not survive frozen validation (Appendix T); it also requires request-time scored probes.

Figure 2 places the four evaluation settings on two axes that organize the transfer regimes observed in our controlled study: how much leverage the injected memory has over the trajectory, and whether its structure is compatible with the request. In that controlled harmful regime, negative transfer concentrates in the high-leverage, structurally incompatible quadrant. The axes describe where we observed harm rather than characterizing it in general: on AppWorld a structurally mismatched workflow was helpful on average (Appendix R).

C ENCODER, CELLS, AND THE COMPUTE LEDGER

The encoder h in full. h is hand-written, deterministic, and reads exactly three things: the request string x, the memory string m, and the static tool schemas $\mathcal{T}$ (tool names and their argument types). It emits a discrete cell. In the notation below, $\textsc{playbookTool}(m)$ is the tool named in the last "Playbook: call `'X'`" position of m; $\textsc{recTools}(m)$ is every backtick-quoted tool mention in m; $\mathrm{obj}(x)$ is the object identifier appearing in the request; and $\mathrm{objOf}(m)$ is the object the playbook line targets.

```
h(x, m, T):
  pb <- playbookTool(m)
  if pb not in T.toolNames:            return ("invalid-tool",)
  if "Both playbooks exist" in m:      return ("conflict",)
  if pb is None or pb is domain-free:  return ("unrelated",)
  for t in recTools(m) \ {pb}:         # memory contradicts itself
      if ("wanted `"+t) in m:          return ("contradict",)
  if obj(x) in m and objOf(m) != obj(x):
                                       return ("partial-arg",)
  prov   <- "same" if obj(x) in m else "cross"   # lexical
  pbAct  <- "write" if T.isWrite(pb) else "read" # static schema
  intent <- "unknown"                  if x is blind
            else "write" if writeVerb(x) else "read"
  am     <- "unknown"                  if intent == "unknown"
            else "match" if pbAct == intent else "mismatch"
  return (prov, am)
```

Every feature is request-time observable. Table 4 states, for each feature, what it is computed from and what it is *not*. The point of the table is that no branch consults the ground-truth action, the task label, the memory-type name, an execution outcome, or a transfer measurement. In particular *provenance* is not an experimenter label: it is the lexical test $\mathrm{obj}(x) \in m$, so a supportive and a corrupted memory about the same object both receive $\mathrm{prov} = $ *same* and are separated instead by the integrity branches, which read the memory's own self-description. On blind requests intent is *unknown* by construction rather than inferred, so the cell records that the request underdetermines the action instead of importing the answer; this is why the certified gate has nothing to certify on the blind slice at the initial evidence level.

Feature	Computed from	Never reads
`invalid-tool`	m, tool names in $\mathcal{T}$	outcome, label
`conflict`	m (self-description)	outcome, label
`unrelated`	$m, \mathcal{T}$	outcome, label
`contradict`	m (states a want, plays another)	outcome, label
`partial-arg`	$\mathrm{obj}(x)$ vs. $\mathrm{objOf}(m)$	outcome, label
`prov`	lexical test $\mathrm{obj}(x) \in m$	memory-type name
`pbAct`	read/write class of `pb` in $\mathcal{T}$	executed tool
`intent`	write-verbs in x; *unknown* if blind	correct action

Table 4: Every branch of h and its inputs. No feature is a function of the ground-truth action, the task label, the experimenter's memory-type name, an execution outcome, or a measured transfer value, so the cell a pair lands in is computable at request time before any rollout.

The probe-augmented encoder h^+, and why it is a separate object. $h^+(x, m, \mathcal{T}, \widehat{U}_0)$ appends to the static cell a band of $\widehat{U}_0$, a request-time estimate of the no-memory return averaged over $R_0{=}4$ *scored* rollouts. Three consequences must be stated rather than glossed. First, h^+ is not a deterministic function of $(x, m, \mathcal{T})$: because $\widehat{U}_0$ is random, the same pair can land in different h^+ cells across probe draws, whereas its static h cell is fixed. All conformal statements for the probe variant are therefore made on the enlarged probability space that includes the probe draws, with the binning (threshold 0.5, $R_0{=}4$) frozen before any calibration residual is computed, so exchangeability holds conditional on the frozen binning. Second, h^+ requires a resettable environment and a request-time outcome scorer; the static encoder requires neither, and it is the static encoder that supports the "no

request-time probes" comparison against CMI. Third, the cost is not amortized: the four probe rollouts recur at every request, so the break-even analysis of Appendix Q applies to TRACE-AVG and not to this variant. Results obtained with h^+ are labelled TRACE-CERT+probe throughout and are never pooled with static-encoder results.

One implementation shortcut, stated. The `partial-arg` branch is implemented by testing the benchmark's second object constant against the playbook line rather than by parsing the object out of the playbook string. We flag it because it is the one place where the released code touches a benchmark-side constant, and we do not ask the reader to take the equivalence on trust. Replacing the branch with a parser that reads the playbook argument out of the memory text and compares it with the request's own object id changes *no* cell assignment on any task we report: 0 of $8{,}448$ (task, kind) pairs across practice, deployment, the practice extension and both confirmatory folds, with the `partial-arg` cell still firing on the same 96 practice pairs. Since the cell is the only channel through which h enters an estimate, every reported statistic is unchanged by construction. The audit ships as a separate script and makes no model calls.

Compute ledger. Table 5 lists every cached track. Records are episode-level: one row per (task, arm, repetition). Baseline (*none*) arms are shared within a task and across memory types, and re-analyses in this paper add no calls because they recompute from these caches. The two $46{,}080$-episode rows are the exploratory certification round (Track G) and the independent frozen-representation validation of it (Track W, Appendix T); they are separate collections on disjoint task ids, and certification therefore accounts for $92{,}160$ of the $146{,}261$ episodes, or about 63%.

Track	Serving	Records
W frozen certification *validation*	hosted API	46,080
G certification round (exploratory)	hosted API	46,080
F cross-family (two executors)	hosted API	6,912
F0 abandoned first attempt (unused)	hosted API	6,912
I local Qwen3-14B extension	local AWQ	8,192
E full-scale controlled benchmark	hosted API	5,376
J size ladder (8B / 14B / 32B)	local AWQ	10,368
O confirmatory fold C (practice)	hosted API	3,456
X confirmatory fold D (deployment)	hosted API	1,536
M BFCL injected	hosted API	3,000
N practice extension (+96 tasks)	hosted API	2,592
T CMI perturbation probes	hosted API	2,304
E LLM-judge baseline	hosted API	768
C main / blind slice	hosted API	792
Q AppWorld (Sonnet; Haiku aborted)	hosted API	359
AA same-app direction pilot	hosted API	195
Z organic pilot, gate + replication	hosted API	120
S organic BFCL multiple + CMI	hosted API	351
D active repair	hosted API	342
R organic BFCL simple	hosted API	270
B pilots (explicit, ambiguous)	hosted API	256
Hosted API total		**127,701**
Local AWQ total		**18,560**
Total		**146,261**

Table 5: Per-track compute ledger, counted directly from the append-only caches. The certification round dominates, which is the honest reading of what the exploratory $13/32$ opening cost, an opening a frozen validation fold later failed to reproduce (Appendix T); it is a separate high-budget study, not a by-product of the main benchmark. The F0 row is an abandoned first cross-family attempt on a model family we did not adopt; those episodes support no reported result and no analysis reads them, but they were spent and are listed rather than dropped.

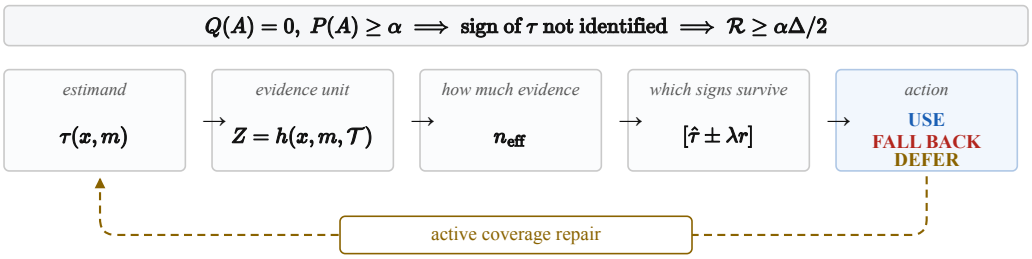


Figure 3: The paper in one strip. The transfer effect is the estimand; the structural representation decides which past pairs count as evidence for a new pair; the effective sample size says how much of that evidence actually supports a given gate; the interval says which signs remain compatible with it; and the action is taken only as strongly as the interval allows. The banner is the constraint that produces the third action: on a region practice never covered, no rule identifies the transfer sign, so a mandatory binary gate is provably lossy. Deferral feeds active coverage repair, which acquires the missing practice rather than guessing through it.

D Claim-to-Experiment Roadmap

Each experiment in §6 is built to check one prediction of §4, including the predictions that are unflattering when they hold.

Claim	**Prediction if true**	**Where tested**
relevance $\neq$ transfer	structure, not similarity, sets the sign	Table 1
the sign is a mechanism	it survives changing the executor	Table 8
scale is not a fix	observed harm grows in the evaluated ladder	§P
zero support $\Rightarrow$ no sign	the sign is unidentified on the hole	Thm. 2, §K
thin evidence $\Rightarrow$ no claim	**the gate defers at this level**	§M
more evidence re-opens it	budget + refined $h^+ \Rightarrow$ *nominal* use	§M
coverage sets claim strength	AVG acts where CERT abstains	Table 2
selector needs enough N	formal rule picks π_0 when N is small	§N
value survives transport	frozen policy keeps a positive LCB on J_D	§O
certificate is marginal	marginal $\not\Rightarrow$ subgroup coverage	§P
scope is conditional	no harmful class $\Rightarrow$ nothing to suppress	§R, Z

The three blocks answer different kinds of question, and conflating them is the mistake this table exists to prevent. Row four is a statement about *identification*: when $Q_Z(A) = 0$ the transfer sign on A is not a function of anything observable before deployment, which is Theorem 2 and is checked against ground truth in the simulator, where a true support hole can be constructed by fiat. Rows five and six are about *finite samples* and are a different phenomenon: in the certification study every cell has some support, and the gate abstains because the radius is large relative to within-cell heterogeneity, not because the sign is unidentified. We therefore do not present the round-0 result ($0/32$ uses, utility equal to the never-memory baseline) as a test of Theorem 2; it is a test of whether the calibrated radius is honest when evidence is thin. Supplying more evidence (R=96 pair means, $\geq$ 39 residuals per cell) together with a refined, probe-augmented representation opened $13/32$ nominal uses; that is exploratory, because the representation was refined after inspecting coverage on the same data, and an independent frozen-representation fold subsequently produced $0/32$ (Appendix T). Row eight is a permission, not a prediction: a marginal guarantee *allows* subgroup coverage to fall below nominal without being violated, so the one slice of 42 that fails ($0.879 < 0.90$) illustrates the gap between marginal and conditional coverage rather than confirming a forecast.

E ASSUMPTIONS AND ESTIMATOR DETAILS

This appendix collects the assumptions, conditional-mean definitions, and the evaluator-reliability lemma on which the theory of §4 rests. The four assumptions correspond to the four ways our estimate of a gate's value can be wrong, and each appears as a named term in the final bound. In words: we can only re-weight practice into deployment if practice actually covers the deployment structures (Overlap); even with perfect data and perfect re-weighting, the conditional transfer law inside a structure may differ between practice and deployment, because h is too coarse to pin it down (Representation mismatch); our evaluator may mislabel outcomes (Evaluator bias); and we must not peek at the same data we calibrate on (Sample splitting).

Definition 3 (Conditional-mean effects, importance weight, and gate value)**.** For a structure $z \in \mathcal{Z}$ write the fiber-average effects

$$\mu_Q(z) = \mathbb{E}_Q[\tau(X,M) \mid h(X,M,\mathcal{T}) = z], \qquad \mu_P(z) = \mathbb{E}_P[\tau(X,M) \mid h(X,M,\mathcal{T}) = z],$$

the true average transfer among practice (Q) and deployment (P) pairs that share structure z, and let $\widetilde{\mu}_Q(z)$ be the quantity the (possibly imperfect) evaluator reports for practice pairs at z. The three differ for two reasons we track explicitly, and it is worth separating them from a third that they are easily confused with. A difference in how *often* each structure occurs, $P_Z \neq Q_Z$, is a between-cell mixture shift and is exactly what the importance weight w corrects. What w cannot correct is $\mu_Q(z) \neq \mu_P(z)$: *within* one structure z, deployment may draw a harder mixture of (x,m) pairs than practice did, so the conditional transfer law is not invariant and h is too coarse. That within-cell gap is what ϵ_{rep} charges for. The evaluator may separately be biased ($\widetilde{\mu}_Q$ vs. μ_Q). The importance weight is $w(z) = \frac{dP_Z}{dQ_Z}(z)$. A gate $g : \mathcal{Z} \to \{0,1\}$ from a finite class $\mathcal{G} \ni g_0 \equiv 0$ (the always-off gate g_0 is always available) has *target value*

$$G_P(g) = \mathbb{E}_P[g(Z)\,\Delta] = \mathbb{E}_P[g(Z)\,\mu_P(Z)],$$

the total expected gain the gate produces at deployment; dividing by the acceptance rate $\mathbb{P}_P(g(Z){=}1)$ turns the total into the average gain among accepted pairs.

Assumption 1 (Overlap)**.** $P_Z \ll Q_Z$ with $w(z) = \frac{dP_Z}{dQ_Z}(z)$ and $B = \|w\|_{\infty,Q_Z} < \infty$. Regions violating this are coverage holes (Theorem 2).

Reading. $w(z)$ is the *importance weight*, "how much more (or less) often does structure z appear at deployment than in practice." We need it because we score gates on practice data but care about deployment; multiplying a practice observation by $w(z)$ makes it count as if it were a deployment observation. B caps how large any single weight can get: a pair seen often at deployment but rarely in practice gets a big weight, and if some deployment structure is *never* practiced ($w = \infty$), no re-weighting can rescue it: that is the coverage hole of Theorem 2.

Assumption 2 (Representation mismatch)**.** $\epsilon_{\text{rep}} = \sup_{g\in\mathcal{G}} \left|\mathbb{E}_P[g(Z)\{\mu_Q(Z) - \mu_P(Z)\}]\right|$.

Assumption 3 (Evaluator bias)**.** $\epsilon_{\text{eval}} = \sup_{g\in\mathcal{G}} \left|\mathbb{E}_P[g(Z)\{\widetilde{\mu}_Q(Z) - \mu_Q(Z)\}]\right|$. Execution verification may justify a *smaller* audited ϵ_{eval}, but does not imply $\epsilon_{\text{eval}} = 0$ unless the evaluator is known to be complete for the utility being measured (incomplete unit tests or missed collateral changes leave a residual). We treat $\epsilon_{\text{rep}}, \epsilon_{\text{eval}}$ as *audited sensitivity terms*, not quantities automatically estimable from unlabeled target pairs.

Assumption 4 (Sample splitting and frozen round)**.** All finite-sample guarantees are stated for a *single frozen acquisition round*: the bank, retriever, encoder h, local estimator $\widehat{\tau}$ and radius $\widehat{r}$, density-ratio estimator, and gate class $\mathcal{G}$ are fixed on folds disjoint from the fold used to estimate gate values, and within that round the practice sample is i.i.d. from a fixed Q. Active repair (§5) collects data adaptively *across* rounds; between rounds we freeze the updated bank and pair law, refit the nuisances, and recalibrate on a fresh held-out fold, so each round's certificate is again single-round valid. We do not claim a single bound over the adaptively pooled mixture $Q_{\text{mix}} = \sum_t \omega_t Q_t$; the closed loop is evaluated empirically.

Evaluator reliability as a bias penalty. The evaluator bias ϵ_{eval} is not assumed away by execution verification; we audit it through a per-observation reliability budget.

Proposition 3 (Reliability as a bias penalty, continuous outcomes)**.** *Let Δ_i^{clean} be the clean paired return difference and $\widetilde{\Delta}_i$ the measured one. If $\eta_i \geq \mathbb{E}[\,|\widetilde{\Delta}_i - \Delta_i^{\text{clean}}| \mid Z_i\,]$ is a reliability error*

budget, then $b_{\mathrm{eval}}(Z_i) \le \eta_i$. *In the special case of discrete corruption where a score* q_i *lower-bounds the probability the label is uncorrupted,* $\eta_i = 2(1 - q_i)$*, since labels differ by at most* 2.

Proof. $|\mathbb{E}[\widetilde{\Delta}_i - \Delta_i^{\mathrm{clean}} \mid Z_i]| \le \mathbb{E}[|\widetilde{\Delta}_i - \Delta_i^{\mathrm{clean}}| \mid Z_i] \le \eta_i$; the discrete case bounds $\mathbb{E}|\cdot| \le 2\mathbb{P}(\widetilde{\Delta}_i \ne \Delta_i^{\mathrm{clean}} \mid Z_i) \le 2(1 - q_i)$. □

The local radius therefore adds $\sum_i \alpha_i(z)\eta_i$ as its evaluator-reliability penalty; reliability enters as bias, never as a reweighting of the estimate. We do not assume execution verification gives $\eta_i = 0$ automatically: incomplete unit-test coverage or missed collateral changes leave a residual η_i that must be audited.

F Uniform Target-Gate Bound

We estimate the deployment value $G_P(g)$ of *every* gate from a single practice sample. The estimator simply re-weights each practice observation by its importance weight:

$$\widetilde{G}_N(g) = \frac{1}{N}\sum_{i=1}^{N} w(Z_i)\, g(Z_i)\, \widetilde{\Delta}_i, \qquad C_2 = \sup_{g\in\mathcal{G}} \mathbb{E}_Q[w(Z)^2 g(Z)].$$

Reading the estimator left to right: keep only accepted pairs ($g(Z_i)$=1), take their measured effect $\widetilde{\Delta}_i$, and weight by $w(Z_i)$ so the average reflects deployment rather than practice. The coefficient C_2 is the *second-moment* (variance-like) size of those weights, it controls how noisy the re-weighted average is. It replaces $\|w\|_\infty$ in the leading variance term, though $B = \|w\|_\infty$ still enters the $1/N$ tail term, so this is a refinement of, not an escape from, the worst-case $\|w\|_\infty$: a few large weights inflate variance through C_2 rather than dominating the whole bound.

F.1 Proof of Theorem 1 (uniform target-gate bound)

Restated. Let $t_\delta = \log(2|\mathcal{G}|/\delta)$. Under Assumptions 1–4, with probability $\ge 1 - \delta$,

$$\sup_{g\in\mathcal{G}} \left|\widetilde{G}_N(g) - G_P(g)\right| \le \underbrace{\sqrt{\frac{2C_2 t_\delta}{N}}}_{\text{(a) noise}} + \underbrace{\frac{4Bt_\delta}{3N}}_{\text{(b) heavy-weight tail}} + \underbrace{\epsilon_{\mathrm{rep}}}_{\text{(c) shift}} + \underbrace{\epsilon_{\mathrm{eval}}}_{\text{(d) evaluator}}.$$

What the four terms mean. The bound says our estimate of every gate's value is off by at most the sum of four interpretable pieces: (a) ordinary statistical noise that shrinks like $1/\sqrt{N}$ as we collect more practice; (b) a smaller $1/N$ term that only matters when a few weights are very large; (c) the distribution-shift gap ϵ_{rep} between practice and deployment structures; and (d) the evaluator bias ϵ_{eval}. Only (a) and (b) shrink with more data; (c) and (d) are the audited budgets from the assumptions, which is exactly why we treat them as sensitivity quantities rather than something we can drive to zero by sampling more. The $\log|\mathcal{G}|$ inside t_δ is the modest price of certifying *all* gates at once (a union bound), so that we may later pick the empirically best one.

Proof. Fix g and set $V_i = w(Z_i)g(Z_i)\widetilde{\Delta}_i$. Since $\widetilde{\Delta}_i \in [-1, 1]$ and $g \in \{0, 1\}$ we have $|V_i| \le B$, $\mathrm{Var}(V_i) \le \mathbb{E}[V_i^2] \le \mathbb{E}_Q[w(Z)^2 g(Z)] \le C_2$ (using $g^2 = g$, $\widetilde{\Delta}^2 \le 1$), and the centered variables $Y_i = V_i - \mathbb{E}V_i$ satisfy $|Y_i| \le 2B$. Bernstein's inequality with this centered range gives

$$\mathbb{P}\left(\left|\tfrac{1}{N}\textstyle\sum_i Y_i\right| > \sqrt{\tfrac{2C_2 t}{N}} + \tfrac{4Bt}{3N}\right) \le 2e^{-t};$$

the linear term carries $\frac{4Bt}{3N}$ (not $\frac{2Bt}{3N}$) precisely because the centered range is $2B$, not B. Set $t = t_\delta$ and union-bound over $\mathcal{G}$. By change of measure $\mathbb{E}_Q[V_i] = \mathbb{E}_P[g(Z)\widetilde{\mu}_Q(Z)]$, so

$$|\mathbb{E}_Q[V_i] - G_P(g)| \le \left|\mathbb{E}_P[g(\widetilde{\mu}_Q - \mu_Q)]\right| + \left|\mathbb{E}_P[g(\mu_Q - \mu_P)]\right| \le \epsilon_{\mathrm{eval}} + \epsilon_{\mathrm{rep}},$$

and combining the stochastic and systematic terms proves the claim. □

Let $\widehat{w}$ be estimated on an independent fold and clipped to $[0, \widehat{B}]$, and define $\widehat{G}_N(g) = \frac{1}{N}\sum_i \widehat{w}(Z_i)g(Z_i)\widetilde{\Delta}_i$, $\widehat{C}_2 = \sup_g \mathbb{E}_Q[\widehat{w}^2 g]$, and $\epsilon_w = \mathbb{E}_Q|\widehat{w}(Z) - w(Z)|$.

Corollary 3 (Sample-split plug-in bound)**.** *Conditional on the nuisance fold, with probability* $\geq 1-\delta$*,*

$$\sup_{g\in\mathcal{G}} \left|\widehat{G}_N(g) - G_P(g)\right| \leq \sqrt{\frac{2\widehat{C}_2 t_\delta}{N}} + \frac{4\widehat{B}t_\delta}{3N} + \epsilon_w + \epsilon_{\text{rep}} + \epsilon_{\text{eval}}.$$

Proof. Apply the Bernstein argument with $\widehat{w}$; the extra expectation error is $|\mathbb{E}_Q[(\widehat{w} - w)g\widetilde{\Delta}]| \leq \mathbb{E}_Q|\widehat{w} - w| = \epsilon_w$. □

Remark 2 (Clipping is biased). If weights are clipped, ϵ_w must include both ratio-estimation error and clipping bias; the clipped estimator is *not* unbiased and must not be described as such.

Remark 3 (Scope: single split, not full cross-fitting). Corollary 3 is proved for a *single* train/calibration split: $\widehat{w}$ is fixed on an independent fold and the bound is conditional on it. The K-fold cross-fitted estimator $\widehat{G}_{\text{CF}}(g) = \frac{1}{K}\sum_k \frac{1}{|I_k|}\sum_{i\in I_k} \widehat{w}^{(-k)}(Z_i)g(Z_i)\widetilde{\Delta}_i$ uses a different nuisance estimate per fold; a valid bound allocates δ/K across the per-fold events (or argues by conditional/martingale concentration). We state the single-split result as our formal guarantee and use cross-fitting only for data efficiency; the δ/K-allocation bound is Proposition 5 in Appendix H.

G Effective Sample Size

Re-weighting has a hidden cost: N practice points re-weighted into deployment are worth *fewer* than N effective points, because the weights are uneven. The next identity makes this precise and gives the quantity practitioners actually report.

Proposition 4 (Exact ESS identity)**.** *For a gate with* $p_g = P_Z(g{=}1) = \mathbb{E}_Q[wg] > 0$ *and* $n_{\text{eff}}(g) = Np_g^2/\mathbb{E}_Q[w^2 g]$*,*

$$\sqrt{\frac{2\,\mathbb{E}_Q[w^2 g]\,t}{N}} = p_g\sqrt{\frac{2t}{n_{\text{eff}}(g)}}.$$

Hence the leading error of the accepted-set conditional mean (after dividing by p_g*) scales as* $n_{\text{eff}}(g)^{-1/2}$*.*

Proof. Substitute the definition of $n_{\text{eff}}(g)$ and simplify. □

Reading. $n_{\text{eff}}(g)$ is the *effective number of practice pairs* the gate's accepted region really buys. If the weights are constant on the accepted region, then $n_{\text{eff}}(g) = N\,Q_Z(g{=}1)$, the expected number of *accepted practice* observations (which equals Np_g only in the no-shift case $P_Z = Q_Z$); when weight instead concentrates on a few pairs, $n_{\text{eff}}(g)$ drops far below that. The identity then says the error of the average-among-accepted behaves like $1/\sqrt{n_{\text{eff}}(g)}$, exactly the familiar "$1/\sqrt{\text{sample size}}$" rule but with the *effective* count. This is why we monitor n_{eff} rather than the raw count, and why a region can be "data-rich" yet still uncertain.

Remark 4 (C_2 versus n_{eff}). C_2 is *not* literally $1/n_{\text{eff}}$: a rare large weight can still inflate C_2. The honest statement is the identity above, mediated by the acceptance mass p_g; C_2 is merely less worst-case than C_∞ and is the variance proxy that actually appears.

H Cross-Fitted Estimator

Proposition 5 (Cross-fitted uniform bound)**.** *Partition the practice sample into folds* $I_1, \ldots, I_K$ *of equal size* N/K *and let* $\widehat{w}^{(-k)}$ *be estimated on the complement of* I_k *with* $\epsilon_w^{(k)} = \mathbb{E}_Q|\widehat{w}^{(-k)}(Z) - w(Z)|$ *and clipping bound* $\widehat{B}$*. Then with probability* $\geq 1-\delta$*,*

$$\sup_{g\in\mathcal{G}} \left|\widehat{G}_{\text{CF}}(g) - G_P(g)\right| \leq \frac{1}{K}\sum_{k=1}^{K}\left[\sqrt{\frac{2\widehat{C}_2\,t_{\delta/K}}{N/K}} + \frac{4\widehat{B}\,t_{\delta/K}}{3\,N/K} + \epsilon_w^{(k)}\right] + \epsilon_{\text{rep}} + \epsilon_{\text{eval}},$$

where $t_{\delta/K} = \log(2|\mathcal{G}|K/\delta)$.

Proof. Condition on the complement of I_k: within fold k the summands are i.i.d. with the fixed nuisance $\widehat{w}^{(-k)}$, so Corollary 3 applies at level δ/K, giving the bracketed bound for the fold-k partial estimator. A union bound over the K fold events (each conditional bound holds with probability $\geq 1 - \delta/K$ regardless of the conditioning value, hence unconditionally) and the triangle inequality over the average of the K partial estimators give the claim; the shift and evaluator biases are common to all folds and enter once. □

Note the cost of cross-fitting in this regime: each fold's deviation term is driven by N/K samples at level δ/K, so the statistical rate is not improved over the single split; the benefit is purely that all N observations contribute to estimation, which matters when N is small. This is why the main text treats cross-fitting as a data-efficiency device, not a sharper guarantee.

I Conformal Certificate: Proofs

This appendix proves the two-level certificate of §4: the pessimistic average-transfer guarantee and its candidate-set lift (the average-value selector), the per-pair conformal certificate (the certified gate), and the coverage-hole lower bound.

I.1 The gate, the pessimistic selector, and the average-transfer guarantee

Gate class. To make $\mathcal{G}$ both finite and tied to the deployed rule, we take the *score-scaling threshold family* $\mathcal{G}_{\mathrm{AVG}} = \{g_\lambda : \lambda \in \Lambda_{\mathrm{AVG}}\} \cup \{g_0\}$ over a finite grid $\Lambda_{\mathrm{AVG}} \subset (0, \infty)$: sub-unit scales *are* admitted here, because the average-transfer theorem below constrains only the selected gate's population value, not per-pair coverage: where

$$g_\lambda(z) = \mathbf{1}\{\widehat{\tau}(z) - \lambda\, s(z) > 0\}$$

for a *generic* z-measurable uncertainty score $s(z) \geq 0$ fixed (with $\widehat{\tau}$) on folds disjoint from the selection fold. Neither Theorem 3 nor Proposition 1 below cares *which* score is used, only that the class is finite and frozen. Two instantiations matter:

$$s_{\mathrm{analytic}}(z) = r(z, \delta) + \beta_\gamma(z), \qquad s_{\mathrm{conformal}}(z) = \widehat{\beta}_\gamma(z) + \rho_R,$$

the analytic radius of Corollary 4 and the deployed conformal radius of Proposition 6; our experiments use $s_{\mathrm{conformal}}$ throughout. Whether $\lambda s(z)$ is additionally a *valid confidence radius*, so that the interval supports per-pair claims: is a separate question, settled only by Propositions 7/2 with $\lambda \in \Lambda_{\mathrm{CERT}} \subset [1, \infty)$. Because s depends only on z, g_λ is a genuine function of z, and the deployed top-k rule (§4.3) uses exactly $L_j^\lambda = \widehat{\tau}(z_j) - \lambda s(z_j)$, $U_j^\lambda = \widehat{\tau}(z_j) + \lambda s(z_j)$, the same score, so choosing λ is the single population-level calibration of the deployed thresholds. The population theorem below analyzes the single-candidate value $G_P(g_\lambda) = \mathbb{E}_P[g_\lambda(Z)\Delta]$. We keep the two grids distinct: Λ_{AVG} feeds the average-value selector, Λ_{CERT} the per-pair certificate.

Operational radius under declared bias budgets. Let $R_N(g)$ be a simultaneous radius with $|\widehat{G}_N(g) - G_P(g)| \leq R_N(g)$ for all $g \in \mathcal{G}$ and $R_N(g_0) = 0$. The radius has a *computable statistical part* and an *audited bias part*; we do not pretend the latter is estimated from data. Since each summand $\widehat{w}(Z_i)g(Z_i)\widetilde{\Delta}_i \in [-\widehat{B}, \widehat{B}]$ has range $2\widehat{B}$, Hoeffding with a union bound over the finite class gives

$$R_N(g) = \underbrace{\widehat{B}\sqrt{\tfrac{2\log(2|\mathcal{G}|/\delta)}{N}}}_{\text{statistical (computable)}} + \underbrace{\bar{\epsilon}_w + \bar{\epsilon}_{\mathrm{rep}} + \bar{\epsilon}_{\mathrm{eval}}}_{\text{declared bias budgets}}.$$

(A tighter gate-wise held-out empirical-Bernstein radius $\sqrt{2\widehat{V}_g \log(2|\mathcal{G}|/\delta)/N} + c\widehat{B}\log(2|\mathcal{G}|/\delta)/N$ may replace the Hoeffding term; we use it only as an experimental ablation to avoid extra proof burden.) Define the *pessimistic* selector

$$\widehat{g} \in \arg\max_{g \in \mathcal{G}} \{\widehat{G}_N(g) - R_N(g)\}.$$

Reading. Rather than picking the gate with the highest *estimated* value (which would chase noise and over-accept), we subtract each gate's error bar $R_N(g)$ and pick the best *worst-case* value, the gate that looks good even after we discount it by everything that could be wrong. Because the do-nothing gate g_0 has value and error bar both zero, the winner can never have a negative worst-case value: when in doubt, the rule falls back to using no memory.

Theorem 3 (Pessimistic average-transfer guarantee)**.** *Let* $\mathcal{E}_{\text{bias}} = \{\epsilon_w \le \bar{\epsilon}_w,\ \epsilon_{\text{rep}} \le \bar{\epsilon_{\text{rep}}},\ \epsilon_{\text{eval}} \le \bar{\epsilon_{\text{eval}}}\}$ *be the event that the declared budgets are valid upper bounds. On the intersection of* $\mathcal{E}_{\text{bias}}$ *with the statistical concentration event,* $G_P(\widehat{g}) \ge 0$*, and for every* $g \in \mathcal{G}$*,*

$$G_P(g) - G_P(\widehat{g}) \le 2R_N(g).$$

Empirically chosen budgets therefore yield a sensitivity analysis*, not an unconditional* $(1-\delta)$ *guarantee.*

Proof. Since g_0 has zero estimate, value, and radius, $\widehat{G}_N(\widehat{g}) - R_N(\widehat{g}) \ge \widehat{G}_N(g_0) - R_N(g_0) = 0$, and the lower confidence bound gives $G_P(\widehat{g}) \ge \widehat{G}_N(\widehat{g}) - R_N(\widehat{g}) \ge 0$. For arbitrary g, optimality gives $\widehat{G}_N(\widehat{g}) - R_N(\widehat{g}) \ge \widehat{G}_N(g) - R_N(g)$; combining with $G_P(g) \le \widehat{G}_N(g) + R_N(g)$ and $G_P(\widehat{g}) \ge \widehat{G}_N(\widehat{g}) - R_N(\widehat{g})$ yields $G_P(g) - G_P(\widehat{g}) \le 2R_N(g)$. □

Remark 5 (What "safe" means here: average, not per-pair). Theorem 3 certifies non-negative *total* incremental value $G_P(\widehat{g}) = \mathbb{E}_P[\widehat{g}(Z)\Delta] \ge 0$; when the accepted region is non-empty this equals a non-negative *average* gain among accepted pairs, and when $\widehat{g} = g_0$ it is simply zero. It does *not* certify $\tau(x, m) \ge 0$ for every accepted pair, nor bound the probability of a harmful or catastrophic individual outcome, those are the job of the local analytic or marginal pair-level certificate (Theorem 4, Proposition 6) and the failure-risk metric of §6. We accordingly call this an average-transfer guarantee, and avoid the word "safe".

Remark 6 (Scope: this calibrates a threshold, it does not certify pairs). The value $G_P(g_\lambda)$ is the average transfer of the *single-candidate* proposal law P. Choosing $\widehat{\lambda}$ by Theorem 3 provides only this single-candidate average-value guarantee; it is *not* a claim that $\widehat{\lambda}$ maximizes the average utility of the deployed top-k policy (the separate candidate-set result below), and it does *not* yield any per-pair claim, since Λ_{AVG} admits $\lambda < 1$, the selected scale may shrink the interval below a confidence set. A per-pair guarantee additionally requires $\widehat{\lambda} \in \Lambda_{\text{CERT}}$ ($\lambda \ge 1$) and one of the two certificate forms below (analytic Proposition 7 or conformal Proposition 2).

I.2 Lifting the guarantee to the deployed candidate-set policy

The deployed object is not a single-pair gate but a policy over candidate sets: given $C = (X, M_1, \ldots, M_k)$, the rule π_λ shortlists, compares LCBs, and returns an action in $\{0, 1, \ldots, k\}$ (no-memory or one candidate). Because the same finite grid indexes this policy class, the argument of Theorem 3 transfers verbatim once the value is defined at the right level. We restate and prove the candidate-set selection proposition (Proposition 1 in the main text).

I.3 Proof of Proposition 1 (pessimistic candidate-set selection)

Restated. Let $C_1, \ldots, C_N \overset{\text{i.i.d.}}{\sim} Q_C$ be practice candidate-set contexts (task plus its k retrieved candidates, under the frozen retriever), and for a policy π let $J(\pi) = \mathbb{E}_{Q_C}\big[U_{\pi(C)} - U_0\big] \in [-1, 1]$ be its paired incremental value, estimated by $\widehat{J}_N(\pi) = \frac{1}{N}\sum_i \big[\widetilde{U}_{\pi(C_i)} - \widetilde{U}_{0,i}\big]$. For $\Pi = \{\pi_\lambda : \lambda \in \Lambda_{\text{AVG}}\} \cup \{\pi_0\}$ with $\pi_0 \equiv$ no-memory ($J(\pi_0) = \widehat{J}_N(\pi_0) = 0$), all maps frozen on disjoint folds, $R_N = \sqrt{2\log(2|\Pi|/\delta)/N}$, and $\widehat{\pi} \in \arg\max_{\pi\in\Pi}\{\widehat{J}_N(\pi) - R_N \mathbf{1}\{\pi \ne \pi_0\}\}$: Write $\widetilde{J}_{Q_C}(\pi) = \mathbb{E}_{Q_C}[\widetilde{U}_{\pi(C)} - \widetilde{U}_0]$ for the *measured* target that $\widehat{J}_N$ actually estimates, and define the candidate-set evaluator bias and transport gap

$$\epsilon^C_{\text{eval}} = \sup_{\pi\in\Pi} \big|\widetilde{J}_{Q_C}(\pi) - J_{Q_C}(\pi)\big|, \qquad \epsilon^C_{\text{rep}} = \sup_{\pi\in\Pi} \big|J_{P_C}(\pi) - J_{Q_C}(\pi)\big|.$$

Then with probability $\ge 1 - \delta$: $\widetilde{J}_{Q_C}(\widehat{\pi}) \ge 0$, $J_{Q_C}(\widehat{\pi}) \ge -\epsilon^C_{\text{eval}}$, $J_{P_C}(\widehat{\pi}) \ge -\epsilon^C_{\text{eval}} - \epsilon^C_{\text{rep}}$, and for all π $J_{Q_C}(\pi) - J_{Q_C}(\widehat{\pi}) \le 2R_N + 2\epsilon^C_{\text{eval}}$ (a further $2\epsilon^C_{\text{rep}}$ under P_C). Under an exact evaluator, $\epsilon^C_{\text{eval}} = 0$ and the first two statements coincide at $J_{Q_C}(\widehat{\pi}) \ge 0$.

Proof. Each summand of $\widehat{J}_N(\pi)$ lies in $[-1,1]$ (range 2) and, by the frozen-folds condition, the summands are i.i.d. with mean $\widetilde{J}_{Q_C}(\pi)$; note this is the *measured* value, since $\widehat{J}_N$ is built from $\widetilde{U}$. Hoeffding plus a union bound over the finite Π therefore gives $\sup_\pi |\widehat{J}_N(\pi) - \widetilde{J}_{Q_C}(\pi)| \le R_N$ with probability $\ge 1-\delta$. On that event the pessimistic argument of Theorem 3 applies verbatim with π_0 in the role of g_0 ($\widetilde{J}_{Q_C}(\pi_0) = \widehat{J}_N(\pi_0) = 0$, radius 0): the selected policy's measured lower bound is ≥ 0, giving $\widetilde{J}_{Q_C}(\widehat{\pi}) \ge 0$, and the oracle inequality in the measured target follows from the same three-step comparison. Passing to the true target is one triangle inequality per statement: $|\widetilde{J}_{Q_C} - J_{Q_C}| \le \epsilon^C_{\text{eval}}$ gives $J_{Q_C}(\widehat{\pi}) \ge -\epsilon^C_{\text{eval}}$ and inflates the oracle gap by $2\epsilon^C_{\text{eval}}$ (once on each side of the comparison); a second triangle inequality with ϵ^C_{rep} carries it to P_C. □

Why this term is stated rather than dropped. $\widehat{J}_N$ is computed from what the evaluator reports, so concentration alone can only certify the measured contrast; equating it with the true contrast is an assumption, not a consequence. For our controlled benchmark the evaluator is exact by construction; execution is deterministic and scored against the ground-truth tool and argument, so $\epsilon^C_{\text{eval}} = 0$ and the headline claim is unaffected. We keep the term visible because it is the assumption a deployment with a surrogate or judge-based scorer would have to audit, and because the same discipline is applied to ϵ_{eval} in Theorem 1.

I.4 Proof of Corollary 1 (empirical-Bernstein selector, fixed-design)

Restated. Let $X^\pi_1, \dots, X^\pi_N \in [-1,1]$ be *independent, not necessarily identically distributed*, with the stratum assignment fixed before data collection, and let $\widehat{V}_\pi$ be the sample variance of the per-context paired values $X^\pi_i = \widetilde{U}_{\pi(C_i)} - \widetilde{U}_{0,i}$. Define the fixed-design target $J_C(\pi) = \frac{1}{N}\sum_i \mathbb{E}[X^\pi_i]$. The *independent* (non-i.i.d.) empirical-Bernstein inequality (Maurer & Pontil, 2009, Theorem 11, which holds for bounded independent, not necessarily identically distributed variables and *upper*-bounds a mean by its empirical average plus a radius), applied to $Y_i = (1 - X^\pi_i)/2 \in [0,1]$ so that a *lower* bound on J_C results, rescaled by 2, and union-bounded over the frozen finite class Π, gives with probability $\ge 1-\delta$ the *one-sided lower* event, simultaneously for all π,

$$J_C(\pi) \;\ge\; \widehat{J}_C(\pi) - R^-_N(\pi), \qquad R^-_N(\pi) = \sqrt{\frac{2\widehat{V}_\pi \,\log(2|\Pi|/\delta)}{N}} + \frac{14\log(2|\Pi|/\delta)}{3(N-1)}.$$

Define the *safety selector* $\widehat{\pi}^- = \arg\max_{\pi\in\Pi}\{\widehat{J}_C(\pi) - R^-_N(\pi)\mathbf{1}\{\pi \ne \pi_0\}\}$ with $R^-_N(\pi_0) := 0$; since the no-memory policy has $J_C(\pi_0) = 0$ known without data, the selected score is ≥ 0 and the one-sided event gives $J_C(\widehat{\pi}^-) \ge 0$. For the oracle inequality, apply the bound to both $(1 \pm X^\pi)/2$ to obtain the *absolute* event $|\widehat{J}_C(\pi) - J_C(\pi)| \le R^\pm_N(\pi)$ with $R^\pm_N$ replacing $\log(2|\Pi|/\delta)$ by $\log(4|\Pi|/\delta)$; the *oracle selector* $\widehat{\pi}^\pm = \arg\max_\pi\{\widehat{J}_C(\pi) - R^\pm_N(\pi)\mathbf{1}\{\pi \ne \pi_0\}\}$ then obeys $J_C(\pi) - J_C(\widehat{\pi}^\pm) \le 2R^\pm_N(\pi)$ for every π.

Proof. Maurer–Pontil's Theorem 11 (empirical Bernstein for bounded independent, non-identically-distributed variables in $[0,1]$) states $\frac{1}{N}\sum \mathbb{E}W_i \le \frac{1}{N}\sum W_i + \sqrt{2\widehat{V}^W \log(2/\delta)/N} + 7\log(2/\delta)/3(N-1)$. Applying it to $W_i = Y_i = (1 - X^\pi_i)/2$ and using $\mathbb{E}[(1-X)/2] = (1-J_C)/2$ gives, after multiplying by 2 and negating, $J_C(\pi) \ge \widehat{J}_C(\pi) - 2r$ with $r = \sqrt{2\widehat{V}^Y \log(2/\delta)/N} + 7\log(2/\delta)/3(N-1)$; since $\widehat{V}^Y = \frac{1}{4}\widehat{V}_\pi$, $2r$ equals $R^-_N(\pi)$ with the $\log(2/\delta)$ from Theorem 11's own constant. A union bound over Π replaces δ by $\delta/|\Pi|$, yielding $\log(2|\Pi|/\delta)$ in the one-sided radius; the two-sided absolute event applies the inequality to both $(1 \pm X)/2$, splitting δ as $\delta/(2|\Pi|)$ per side and yielding $\log(4|\Pi|/\delta)$. Setting $R^-_N(\pi_0) = 0$ is valid because $J_C(\pi_0) = 0$ deterministically. The safety and oracle steps then follow as in Proposition 1, each for its own selector $\widehat{\pi}^-/\widehat{\pi}^\pm$. Under the pre-fixed balanced grid below, J_C is the balanced-population mean; no unconditional i.i.d. draw from Q_C is required. □

Reading. This is the guarantee the deployed average-value selector actually inherits: selecting λ pessimistically over the finite grid yields a top-k policy whose *average* paired gain on the fixed-design

confirmatory target is non-negative ($J_C(\widehat{\pi}^-) \geq 0$), degrading on deployment by at most the fixed-design gap $\epsilon_{\text{rep}}^{C,\text{grid}} = \sup_{\pi\in\Pi} |J_{P_C}(\pi) - J_C(\pi)|$, so that $J_{P_C}(\widehat{\pi}^-) \geq -\epsilon_{\text{rep}}^{C,\text{grid}}$. (When the selection target is instead the i.i.d. practice law Q_C, the corresponding gap is the ϵ_{rep}^C of Proposition 1; the two coincide if the balanced grid matches the Q_C mixture.) We treat this gap as a declared *sensitivity parameter*, not an estimated quantity: the practice-vs-deployment cell-mean agreement we report is supporting evidence that the underlying cell effects transport, but it is *not* a valid upper bound on ϵ_{rep}^C, which also absorbs candidate-selection effects and cross-candidate correlations. No importance weighting is needed for the practice-side statement because the value is estimated by direct paired rollouts on practice contexts; the coverage machinery of Theorem 1 is only needed when practice outcomes must be *re-weighted* toward a different pair law. Per-pair claims still belong exclusively to Proposition 2 with $\lambda \in \Lambda_{\text{CERT}}$.

I.5 Cell-mean and marginal pair-mean certificates

The population gate answers "on average, is this gate's accepted region helped?" For the actual deployment decision we need the sharper, local question: "for *this* request and *this* memory, is the effect positive?" We answer it by a local average of nearby practice pairs, a nearest-neighbour / kernel estimate in structural-pair space. With local weights $\alpha_i(z) \geq 0$, $\sum_i \alpha_i(z) = 1$ (larger for practice pairs Z_i structurally closer to z), set $\widehat{\tau}(z) = \sum_i \alpha_i(z)\widetilde{\Delta}_i$ and $n_{\text{eff}}(z) = 1/\sum_i \alpha_i(z)^2$. Here we do *not* importance-weight: we are estimating a local effect, and re-weighting would only add variance. We assume the true effect varies smoothly across structures, $|\mu_Q(z) - \mu_Q(z')| \leq L\, d(z, z')$, and write $b_{\text{rep}}(z) = |\mu_P(z) - \mu_Q(z)|$ for the practice-vs-deployment gap and $b_{\text{eval}}(z) = |\widetilde{\mu}_Q(z) - \mu_Q(z)|$ for the evaluator bias, both at this z.

A subtlety the representation forces: fiber heterogeneity. Because the encoder h is not injective, $\mu_P(z) = \mathbb{E}_P[\tau(X, M) \mid h(X, M, \mathcal{T}) = z]$ is the *average* effect over all pairs that share structure z (the *fiber* of z), not the effect of the one pair (x, m) in front of us. If two pairs with the same z have effects $+1$ and -1 in a $60/40$ mix, then $\mu_P(z) = 0.2 > 0$ even though the current pair may truly have effect -1. No amount of data, perfect overlap, or zero evaluator bias removes this gap; it is intrinsic to summarizing a pair by a coarser z. We name it explicitly,

$$b_{\text{fiber}}(x, m) = \big|\tau(x, m) - \mu_P(h(x, m, \mathcal{T}))\big|,$$

the deviation of the true pair effect from its fiber average, which vanishes exactly when h is a sufficient statistic for transfer, $\tau(x, m) = f(h(x, m, \mathcal{T}))$; an assumption we do *not* make (a richer h shrinks it, a coarser one inflates it).

Two ways to carry the certificate to the pair. A *deterministic* per-pair certificate needs a valid pointwise bound $\bar{b}_{\text{fiber}}(x, m) \geq b_{\text{fiber}}(x, m)$. This is honest but operationally demanding: the only universal bound, $b_{\text{fiber}} \leq 2$, is vacuous. We therefore prefer a *z-measurable, probabilistic* envelope: a within-fiber quantile

$$\beta_\gamma(z) = Q_{1-\gamma}\big(|\tau(X, M) - \mu_P(z)| \;\big|\; Z = z\big), \qquad \mathbb{P}\big(|\tau(X, M) - \mu_P(Z)| \leq \beta_\gamma(Z)\big) \geq 1 - \gamma,$$

which depends only on z and is auditable from held-out *pair means*. We make the z-level statement first, then add either form to reach the pair.

Theorem 4 (Fixed-structure conditional-mean certificate)**.** *This certifies the fiber average $\mu_P(z)$ at a fixed structure z (the per-pair extension is Corollary 4). Let the index i run over independent* task–candidate–model units*. Only repeated stochastic rollouts of the* same *unit are averaged into $\widetilde{\Delta}_i = \frac{1}{R}\sum_r \Delta_{i,r}$; different memories or model conditions are* never *averaged, since they define different structural pairs and different transfer effects. Conditional on the structural pairs, for any fixed z, with probability $\geq 1 - \delta$,*

$$|\widehat{\tau}(z) - \mu_P(z)| \leq L \sum_i \alpha_i(z)\, d(Z_i, z) + \sum_i \alpha_i(z)\, b_{\text{eval}}(Z_i) + b_{\text{rep}}(z) + \sqrt{\frac{2\log(2/\delta)}{n_{\text{eff}}(z)}}.$$

Proof. Decompose $\widehat{\tau}(z) - \mu_P(z)$ into $\sum_i \alpha_i\{\widetilde{\Delta}_i - \widetilde{\mu}_Q(Z_i)\}$, $\sum_i \alpha_i\{\widetilde{\mu}_Q(Z_i) - \mu_Q(Z_i)\}$, $\sum_i \alpha_i\{\mu_Q(Z_i) - \mu_Q(z)\}$, and $\mu_Q(z) - \mu_P(z)$. The last three are bounded by the evaluator-bias,

Lipschitz, and representation terms. Conditional on $Z_{1:N}$ the first is a sum of independent mean-zero variables with interval lengths $\le 2\alpha_i$, so Hoeffding gives $\mathbb{P}(|\cdot| \ge t \mid Z_{1:N}) \le 2\exp(-t^2/2\sum_i \alpha_i^2)$; set the bound to δ. □

The four terms, in words. The local estimate can be wrong for exactly four reasons, and the bound is just their sum: (1) $L\sum_i \alpha_i\, d(Z_i, z)$; neighbours aren't exactly z; (2) $\sum_i \alpha_i b_{\mathrm{eval}}(Z_i)$, the evaluator mislabels the neighbours; (3) $b_{\mathrm{rep}}(z)$: practice and deployment effects at z differ; and (4) $\sqrt{2\log(2/\delta)/n_{\mathrm{eff}}(z)}$: ordinary sampling noise, which shrinks only as $n_{\mathrm{eff}}(z)$ grows. Only term (4) is reduced by collecting more practice near z; terms (1)–(3) are bias terms that improve only with a better representation or evaluator.

Corollary 4 (Per-pair certificate: deterministic and probabilistic)**.** *Let $z = h(x, m, \mathcal{T})$ and let $r(z, \delta)$ be the radius of Theorem 4.* (a) Deterministic. *On the Theorem 4 event, if $\bar{b}_{\mathrm{fiber}}(x, m) \ge b_{\mathrm{fiber}}(x, m)$ is a valid bound, then $|\widehat{\tau}(z) - \tau(x, m)| \le r(z, \delta) + \bar{b}_{\mathrm{fiber}}(x, m)$.* (b) Probabilistic, z-measurable. *For a* random *deployment pair (X, M) with $Z = z$, using the within-fiber envelope $\beta_\gamma(z)$,*

$$\mathbb{P}\Big(|\widehat{\tau}(Z) - \tau(X, M)| \le r(Z, \delta) + \beta_\gamma(Z)\Big) \;\ge\; 1 - \delta - \gamma.$$

Proof. (a) Triangle inequality: $|\widehat{\tau}(z) - \tau(x, m)| \le |\widehat{\tau}(z) - \mu_P(z)| + b_{\mathrm{fiber}}(x, m) \le r(z, \delta) + \bar{b}_{\mathrm{fiber}}(x, m)$. (b) Combine the Theorem 4 event (probability $\ge 1 - \delta$) with the envelope event $|\tau(X, M) - \mu_P(Z)| \le \beta_\gamma(Z)$ (probability $\ge 1 - \gamma$) via a union bound. □

Reading. The pair radius is the fiber-average radius *plus* how much this pair can differ from others sharing its structure. Form (a) gives a worst-case guarantee for every pair but needs a pointwise bound that is hard to obtain non-trivially; form (b) gives an *auditable* guarantee for a random deployment pair using only the z-measurable envelope $\beta_\gamma(z)$. In practice $r^{\mathrm{pair}}(z)$ is the conformal radius $\widehat{\beta}_\gamma(z) + \rho_R$ centered directly at $\widehat{\tau}(z)$ (Proposition 6), which already bounds $|\tau - \widehat{\tau}|$ and so subsumes both the smoothness radius and the fiber term rather than adding them.

I.6 Calibrating the envelope with finite data

Corollary 4(b) assumes the *population* envelope $\beta_\gamma(z)$. In practice we calibrate an empirical $\widehat{\beta}$ by split-conformal quantiles of fiber residuals, valid at finite sample under exchangeability; we state it for a discrete structural cell (Mondrian conformal), the continuous case following by cell-grouping or weighted conformal.

What one calibration unit is. Exchangeability is a statement about the calibration *units*, so it matters that a unit is not silently a within-task replicate. In our protocol a unit is one (task, memory-kind) pair, summarised by its repeated-rollout mean $\bar{\Delta}_i$; the R rollouts inside a unit are aggregated, never entered separately. Crucially, **at most one residual per task enters any given cell**: the arms a task contributes are mapped to cells by h, and distinct arms of the same task land in distinct cells (a same-object arm and a cross-object arm cannot share a cell by construction). We verified this on the certification round rather than assuming it: all 256 calibration units are distinct (task, cell) combinations, i.e. the maximum number of residuals a single task contributes to a single cell is exactly 1. Calibration units within a cell are therefore across-task, so shared task-level difficulty cannot induce within-cell dependence. We stress what this does and does not buy: removing within-task clustering is necessary for ordinary split-conformal exchangeability to be the right object, but it is not sufficient for it. Exchangeability additionally requires the calibration and deployment units inside that cell to follow the same residual law, which is the transport assumption stated after Proposition 6 and audited rather than proved. Had a task been able to contribute several residuals to one cell, a task-clustered conformal formulation would have been required on top of that. Consistently, all reported confidence intervals resample *tasks* as the cluster unit, with candidate memories and seeds treated as within-cluster.

Proposition 6 (Conformal per-pair envelope)**.** *Center the conformal residuals at the* deployed estimator $\widehat{\tau}(z)$ *itself (fixed on a training fold, disjoint from calibration), so no separate $\widehat{\mu}_P$ enters. Fix a structural cell z; let (X_i, M_i), $i \in \mathcal{C}_z$ ($|\mathcal{C}_z| = n_z$), be calibration pairs with $h = z$, each with a*

direct repeated-rollout mean $\bar{\Delta}_i$ *over* R *rollouts and residual* $R_i = |\bar{\Delta}_i - \widehat{\tau}(z)|$*, exchangeable with a fresh deployment pair* (X, M) *at* z*. Set*

$$\widehat{\beta}_\gamma(z) = \lceil (n_z{+}1)(1-\gamma) \rceil \text{-th smallest of } \{R_1, \ldots, R_{n_z}, +\infty\},$$

with the convention that this value is $+\infty$ *when the index exceeds* n_z *(too little calibration data to certify, so the interval is vacuous and the system defers). Then*

$$\mathbb{P}\big(|\bar{\Delta}_{(X,M)} - \widehat{\tau}(z)| \le \widehat{\beta}_\gamma(z)\big) \ge 1 - \gamma,$$

and, accounting for the finite-rollout error of the means with the correct *Hoeffding radius for* $\Delta \in [-1, 1]$ *(range* 2*),*

$$\rho_R = \sqrt{\frac{2\log(2/\gamma')}{R}}, \qquad \mathbb{P}\big(|\tau(X, M) - \widehat{\tau}(z)| \le \widehat{\beta}_\gamma(z) + \rho_R\big) \ge 1 - \gamma - \gamma'.$$

Proof. By exchangeability of the $n_z{+}1$ residuals (calibration plus test), the rank of the test residual is sub-uniform on $\{1, \ldots, n_z{+}1\}$ (exactly uniform under no ties or randomized tie-breaking; conservative otherwise), so it falls at or below the $\lceil (n_z{+}1)(1-\gamma) \rceil$-th order statistic with probability $\ge 1 - \gamma$; the $+\infty$ convention covers the index-$> n_z$ edge case. For the rollout term, $\bar{\Delta} = \frac{1}{R}\sum_r \Delta_r$ with $\Delta_r \in [-1, 1]$ (range 2) gives $\mathbb{P}(|\bar{\Delta} - \tau| \ge t) \le 2\exp(-Rt^2/2)$; setting the bound to γ' yields $\rho_R = \sqrt{2\log(2/\gamma')/R}$. A union bound combines the two events. □

The exchangeability premise is an assumption, not a theorem. The proposition's guarantee is conditional on the calibration residuals at cell z being exchangeable with the deployment pair's residual at the same cell. When calibration pairs are drawn from practice (Q) and deployment pairs from P, this is a *residual-transport* assumption: within a structural cell, the distribution of $|\bar{\Delta} - \widehat{\tau}(z)|$ carries over from practice to deployment. It is weaker than full distributional invariance (only the residual law within each Mondrian cell must transport) and is motivated by Assumption 2, but it can fail if deployment concentrates on a harder sub-population inside a cell; §6 and Appendix P audit it directly; transport holds on all full-population slices, and an adversarial sub-population sweep locates exactly one violation, an audit, not a proof.

Consequence. Because the residuals are centered at $\widehat{\tau}(z)$, the proposition certifies $|\tau(X, M) - \widehat{\tau}(z)|$ *directly*: the deployed per-pair radius is $r^{\text{pair}}(z) = \widehat{\beta}_\gamma(z) + \rho_R$, and one does *not* re-add the Theorem 4 radius $r(z, \delta)$ (doing so would double-count uncertainty). This is the conformal instantiation of Corollary 4(b); in the top-k rule both the envelope and the rollout radius are taken at level $\alpha/(2k)$ (Proposition 2).

Remark 7 (Three distinct targets: cell mean, pair mean, next rollout). The deployable forms of the interval certify *different* quantities and must not be conflated. *(a)* $C_{\text{cell-mean}}(z)$*: analytic interval for the cell conditional mean* $\mu_P(z)$. The Lipschitz constant L and the biases $b_{\text{rep}}(z), b_{\text{eval}}(z)$ are not known a priori, so the radius of Theorem 4 is a genuine certificate for $\mu_P(z)$ *only* given valid upper bounds; we deploy it as a *sensitivity* object. *(b)* $C_{\text{pair-mean}}(z)$*: conformal interval for the pair mean effect* $\tau(X, M)$. Calibrating on residuals of repeated-rollout pair means $\bar{\Delta}_i$ (Proposition 6) and adding the rollout radius ρ_R yields marginal $1 - \gamma - \gamma'$ coverage of the pair-level mean effect for an exchangeable deployment pair; outcome noise enters only through ρ_R. *(c)* $C_{\text{next-rollout}}(z)$*: conformal predictive interval for a single future outcome.* A split-conformal quantile of *raw* residuals covers a future paired outcome at level $1 - \delta$; large outcome noise widens it without certifying any mean's sign. The gate's per-pair claims use $C_{\text{pair-mean}}$; $C_{\text{cell-mean}}$ is the worst-case sensitivity statement; $C_{\text{next-rollout}}$ is a rollout-level risk diagnostic, never a decision object.

Remark 8 (How the budgets are audited, and the conditional nature of the guarantee). Theorem 4 holds *conditional on the supplied budgets being valid upper bounds*; empirically selected budgets yield a sensitivity analysis. We audit each on held-out data: the smoothness budget $\bar{L}$ from a high quantile of empirical slopes; the evaluator budget $\bar{b}_{\text{eval}}$ from execution-vs-human disagreement, repeated state-check discrepancy, and missed-collateral-change rate. The hardest term is $\bar{b}_{\text{rep}} = |\mu_P(z) - \mu_Q(z)|$: the oracle-vs-predicted encoder gap measures encoder *error*, and even with identical encoders a harder deployment mixture inside a cell makes $\mu_P(z) \ne \mu_Q(z)$. A valid bound requires direct target signal, a small labeled target paired-rollout audit set, a worst-group confidence bound, an explicit conditional-invariance assumption, or a distributionally robust envelope. Absent these we report $\bar{b}_{\text{rep}}$ as a swept sensitivity budget, not an estimated upper bound.

I.7 THE DEPLOYED DECISION: TOP-k SELECTION WITH A JOINT CERTIFICATE

A retriever returns candidates $m_1, \dots, m_k$, encoded as pairs $z_j = h(x, m_j, \mathcal{T})$, so the deployment action set is $\mathcal{A}_x = \{0, m_1, \dots, m_k\}$. Selecting the empirically best candidate is an outcome-dependent comparison, so the certificate must hold *simultaneously* over the k candidates. Given per-candidate radii r_j and a scale $\lambda \geq 1$, compute $[L_j^\lambda, U_j^\lambda] = [\widehat{\tau}(z_j) - \lambda r_j,\ \widehat{\tau}(z_j) + \lambda r_j]$, let $j^\star = \arg\max_{j \in [k]} L_j^\lambda$, and act by

$$\pi_\lambda(x, m_{1:k}) = \begin{cases} \text{use } m_{j^\star}, & L_{j^\star}^\lambda > 0, \\ \text{fallback (no memory)}, & \max_{j \in [k]} U_j^\lambda < 0, \\ \text{defer}, & \text{otherwise.} \end{cases}$$

The scale must not shrink the radius. If r_j is a valid radius, any $\lambda \geq 1$ preserves coverage; any $\lambda < 1$ *destroys* it; the interval is no longer a confidence set. Certified deployment therefore requires $\lambda \in \Lambda_{\mathrm{CERT}} \subset [1, \infty)$; sub-unit scales are legitimate only for the average-value regime of Theorem 3 and Proposition 1.

Proposition 7 (Joint top-k certificate, analytic form)**.** *Fix $\lambda \geq 1$ and let each radius be the analytic per-pair radius of Corollary 4, $r_j = r(z_j, \delta/k) + \beta_{\gamma/k}(z_j)$. Let $\mathcal{E}_{\mathrm{budget}}$ be the event that the declared budgets are valid upper bounds. Then on $\mathcal{E}_{\mathrm{budget}}$, simultaneously for all j, $\tau(x, m_j) \in [L_j^\lambda, U_j^\lambda]$ with probability $\geq 1 - \delta - \gamma$. Consequently the rule's three branches carry their stated meanings:* use *certifies $\tau(x, m_{j^\star}) \geq L_{j^\star}^\lambda > 0$ despite the data-dependent selection;* fallback *certifies $\tau(x, m_j) < 0$ for every j;* defer *means no candidate is certified positive and at least one is sign-unresolved.*

Proof. Each candidate has a statistical failure (Theorem 4 at level δ/k) and a fiber-envelope failure (population envelope $\beta_{\gamma/k}$; under the uniform proposal the mixture bound yields $\sum_j \mathbb{P}(F_j) \leq \gamma$). A union bound over both modes and all k candidates gives total failure $\leq \delta + \gamma$; on the complement, $\lambda \geq 1$ preserves each interval, and the three branch implications follow as in the single-pair case. □

I.8 PROOF OF PROPOSITION 2 (JOINT TOP-k CERTIFICATE, CONFORMAL FORM)

Restated. Fix $\lambda \geq 1$ and a total error level α. Let $\widehat{\tau}$ be frozen on a training fold, let the shortlist map be frozen before calibration, and assume calibration pairs and the deployment pair are exchangeable within each cell after the same shortlist map. Allocate the budget as $\gamma_c = \gamma_r = \alpha/(2k)$ and set

$$r_j \ = \ \widehat{\beta}_{\alpha/(2k)}(z_j) \ + \ \sqrt{\tfrac{2\log(4k/\alpha)}{R}}.$$

Then simultaneously for all j, $|\tau(x, m_j) - \widehat{\tau}(z_j)| \leq \lambda r_j$ with probability $\geq 1 - \alpha$, and the three branches carry the same meanings as in Proposition 7. No Theorem 4 event and no $(\bar{L}, \bar{b}_{\mathrm{rep}}, \bar{b}_{\mathrm{eval}})$ budgets enter: the conformal residuals are centered at $\widehat{\tau}$ and already control $|\tau - \widehat{\tau}|$ directly; adding the analytic radius would double-count.

Proof. Each candidate has a conformal failure (the calibrated envelope at level $\alpha/(2k)$, Proposition 6) and a rollout-mean failure ($|\bar{\Delta} - \tau| > \rho_R$, probability $\leq \alpha/(2k)$ by Hoeffding with range 2). A union bound over the $2k$ events gives total failure $\leq k \cdot \alpha/(2k) + k \cdot \alpha/(2k) = \alpha$. On the complement, $\lambda \geq 1$ preserves each interval and the branch implications follow. Omitting the rollout allocation: using ρ_R at a level γ' not split across candidates; would degrade the joint level to $\alpha/2 + k\gamma'$, which is why both budgets are split. □

Remark 9 (the certified gate vs. the average-value selector: two regimes, two names). The propositions above define **Certified gate**: $\lambda \geq 1$, radii from exactly one of the two forms, and only this regime may be described as "certified." Separately, **Avg. selector** chooses *any* $\lambda \in \Lambda_{\mathrm{AVG}}$ (including $\lambda < 1$) by pessimistic maximization; the resulting top-k policy carries the candidate-set *average*-transfer guarantee $J(\widehat{\pi}) \geq 0$ of Proposition 1 (and the single-candidate $G_P(\widehat{g}) \geq 0$ of Theorem 3): provided the simultaneous radius used in the selection is itself theorem-backed (the union-bound Hoeffding radius; a task-cluster bootstrap radius is an *empirical* selector outside the formal guarantee), but *no* per-pair claim. The two regimes trade coverage of the upside against the strength of the per-decision claim, and we keep their names and grids (Λ_{AVG} vs. Λ_{CERT}) distinct throughout.

Interpretation: marginal, not conditional. The conformal guarantee of Proposition 2 is *marginal* over a random deployment request and its retrieved candidate set drawn under the frozen proposal law: with probability $\geq 1-\alpha$ the simultaneous top-k intervals cover. It is *not* a statement that this particular already-observed candidate has positive transfer with that probability *conditional* on (x, m); conditional coverage would need the deterministic form (a) with a valid pointwise $\bar{b}_{\text{fiber}}$. If a shortlist of $K < k$ candidates is used, the shortlist map must be frozen on the training fold, the calibration pairs must pass through the *same* map, and all budgets split over K; if candidate rank affects the residual law, rank must be included in z or the envelope calibrated rank-conditionally.

I.9 Coverage-hole lower bound

Everything so far assumed overlap. This subsection shows the assumption is not cosmetic: where practice gives *no* information about a deployment structure, *no* method can decide correctly. We measure quality by the memory-decision regret against the oracle gate $g^\star_M(z) = \mathbf{1}\{\mu_{P,M}(z) > 0\}$,

$$\mathcal{R}_M(g) = \mathbb{E}_{P_Z}\big[\max\{0, \mu_{P,M}(Z)\} - g(Z)\mu_{P,M}(Z)\big],$$

which adds up, over deployment pairs, the gain the oracle captures but g misses.

I.10 Proof of Theorem 2 (coverage-hole minimax impossibility)

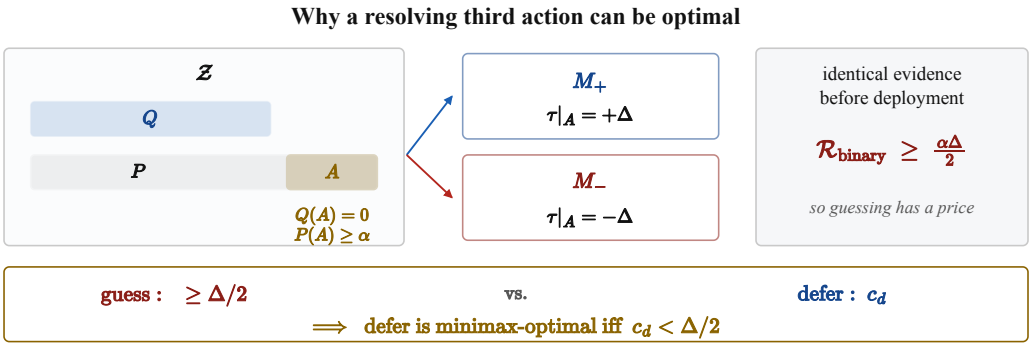


Figure 4: Why a resolving third action can be minimax-optimal. If a region A of structural space carries deployment mass but was never practiced, then the two environments shown agree on everything the agent could have observed before deployment and differ only in the sign of the effect on A. No data-dependent rule can tell them apart, so every mandatory binary gate suffers the stated minimax regret, where α lower-bounds the deployment mass p_A of the hole and Δ is the cost of the wrong sign. Extending the action set with a *defer* that resolves the request at price c_d turns the comparison into two prices: deferral strictly dominates when $c_d < \Delta/2$, ties at equality, and is dominated above it.

Restated. Fix a gap $\Delta \in (0, 1]$ and a *mandatory binary* gate $\widehat{g} \in \{0, 1\}$ (use/fallback, no free defer). Suppose a measurable $A \subseteq \mathcal{Z}$ has $P_Z(A) \geq \alpha$ and $Q_Z(A) = 0$. Consider two environments M_+, M_- that agree outside A, satisfy on A $\mu_{P,M_+}(z) = +\Delta$, $\mu_{P,M_-}(z) = -\Delta$, and under which *the joint law of all pre-deployment information available to the learner*: paired-practice data, tool schemas, request and memory metadata, encoder outputs, density-ratio and auxiliary models, and acquisition history: is identical. Then every data-dependent, possibly randomized gate obeys

$$\inf_{\widehat{g}} \sup_{M \in \{M_+, M_-\}} \mathbb{E}_M[\mathcal{R}_M(\widehat{g})] \geq \frac{\alpha\Delta}{2}.$$

Proof. All pre-deployment information has identical law under the two environments, so the conditional distribution of $\widehat{g}$ on A is the same; let $p(z) = \mathbb{E}[\widehat{g}(z) \mid \text{data}]$. On A, regret is $\Delta(1-p(z))$ under M_+ and $\Delta p(z)$ under M_-, whose average is $\Delta/2$. Integrating over A gives average regret $\geq \alpha\Delta/2$; the max of the two is at least their average, and taking $\inf$ over gates completes the proof. □

The idea in one sentence. We build two worlds that look *identical* in all the data the learner ever sees, but in which the same memory helps in one world ($+\Delta$) and hurts in the other ($-\Delta$) on the un-practiced region A. Any fixed decision is therefore right in one world and wrong in the other, so its worst-case regret is at least half the gap times how often A occurs ($\alpha\Delta/2$). This is why coverage is a genuine prerequisite, and why the active-repair loop exists to fill such holes before they bite.

Remark 10 (Scope: forced decision and admissibility). The theorem is stated over the *unrestricted* conditional-mean class and for a *forced* binary gate: the third action, *defer*, escapes the bound precisely by *not* deciding under a hole, but only at the cost of the deferral mechanism (verification, human, or acquisition), a free, perfect defer oracle would make the bound inapplicable. To keep the same conclusion *inside* the L-Lipschitz class of Theorem 4, take an unobserved hole H with $Q_Z(H) = 0$, a target region $A \subseteq H$ with $P_Z(A) \geq \alpha$ and a buffer $\mathrm{dist}(A, H^c) \geq \Delta/L$, and let M_+, M_- agree on H^c while interpolating L-Lipschitz-ly to $\pm\Delta$ on A; the regret is then evaluated on A and the bound $\alpha\Delta/2$ holds verbatim.

I.11 Proof of Corollary 2 (when deferral is minimax-optimal)

Restated. Extend the action set with *defer*, charged $c_d \geq 0$ per deferred pair and assumed to resolve that pair. Write $p_A = P_Z(A) \geq \alpha$. Under the hypotheses of Theorem 2, deferring on the hole is minimax-optimal whenever $c_d \leq \Delta/2$, and strictly dominates every mandatory binary gate when $c_d < \Delta/2$, ties the binary minimax optimum at $c_d = \Delta/2$, and is dominated for $c_d > \Delta/2$.

Proof. Decompose the regret over A and A^c. Outside A the two environments $M_\pm$ agree and the action sets coincide, so any rule attains the same value there and the comparison reduces to A.

On A, consider first a rule that must choose use or fallback. By Theorem 2 its acceptance probability on A has identical law under M_+ and M_-, since all pre-deployment information does. Write $p = \mathbb{P}(\widehat{g} = 1 \mid Z \in A)$. Against M_+ (where $\mu = +\Delta$) rejecting costs Δ, so the conditional regret is $(1-p)\Delta$; against M_- (where $\mu = -\Delta$) accepting costs Δ, giving $p\Delta$. The worst case is $\max\{p, 1-p\}\Delta \geq \Delta/2$, attained at $p = 1/2$, so the contribution to worst-case regret is at least $P_Z(A)\,\Delta/2 \geq \alpha\Delta/2$.

Now consider deferring on all of A. Deferral does not decide, so it incurs no sign error in either environment; by definition it is charged c_d per pair, and its contribution is exactly $P_Z(A)\,c_d$. Comparing the two,

$$P_Z(A)\,c_d \;<\; P_Z(A)\,\Delta/2 \quad\Longleftrightarrow\quad c_d < \Delta/2,$$

so under that condition the deferring rule has strictly smaller worst-case regret than *any* mandatory binary gate, and no rule that decides on A can match it. Mixed rules that defer on a sub-region $A' \subseteq A$ and decide on $A \setminus A'$ interpolate between the two costs, so the all-defer rule is optimal on A under the same condition. □

Reading. The corollary is what upgrades "abstention is permitted" to "abstention is optimal". It also makes the price explicit: deferral is not free, and if the resolution channel costs more than half the effect size it protects against ($c_d \geq \Delta/2$), guessing is no worse in the minimax sense. Active repair is a *sequential analogue*, not an instance of this corollary. The cost c_d prices a channel that resolves the *current* pair; active repair instead runs the current request without memory and buys evidence that may reduce uncertainty for *future* requests sharing that structure. It therefore does not instantiate the resolving deferral Corollary 2 assumes, and in our experiments acquisition did not in the end open a pair-level certificate (Appendix T). What the corollary establishes is conditional: *if* a resolving channel exists at cost below $\Delta/2$, abstention is not merely permitted but optimal.

J Active Repair: Submodularity

Deferral is an acquisition signal, not a terminal state. A cluster of pairs z_j is worth practicing when it is frequent at deployment, unresolved (its interval straddles zero), and wide:

$$A_j = \widehat{p}_{\mathcal{D}}(z_j)\,\mathbf{1}\{L_j^\lambda \leq 0 \leq U_j^\lambda\}\,(U_j^\lambda - L_j^\lambda)\,\mathrm{Feasibility}(z_j),$$

where Feasibility is the chance a synthetic variant is executable and checkable. We isolate a clean surrogate objective and prove only that it is optimized near-optimally; the full adaptive loop is

evaluated empirically. Partition the target pair space into clusters $j = 1, \ldots, J$ with target masses p_j, constants $c_j \geq 0$, and initial evidence $n_j^0 \geq 0$; a candidate practice item e adds a *known, deterministic* increment $a_{ej} \geq 0$ to cluster j. With a pseudocount $\nu > 0$ set $u_j(n) = c_j/\sqrt{n+\nu}$ and define

$$F(S) = \sum_j p_j \Big[u_j(n_j^0) - u_j\big(n_j^0 + \textstyle\sum_{e \in S} a_{ej}\big) \Big].$$

Reading. $F(S)$ measures how much total (deployment-weighted) uncertainty a chosen practice set S removes; $u_j(n) = c_j/\sqrt{n+\nu}$ is the per-cluster uncertainty, falling as evidence accumulates. The pseudocount ν is a *ranking regularizer* that keeps the objective finite at a never-practiced cluster; it does *not* manufacture a certificate, a true coverage hole still has an unresolved (effectively infinite) certificate radius until real practice arrives.

Proposition 8 (Submodular coverage repair)**.** *F is normalized, monotone, and submodular, so under a budget $|S| \leq B$ greedy selection satisfies $F(S_{\text{greedy}}) \geq (1 - 1/e) \max_{|S| \leq B} F(S)$. The same holds for the binary support-coverage reward $F_{\text{hole}}(S) = \sum_j p_j \, \mathbf{1}\{\sum_{e \in S} a_{ej} > 0\}$.*

Proof. For cluster j, $f_j(t) = c_j/\sqrt{n_j^0 + \nu} - c_j/\sqrt{n_j^0 + \nu + t}$ has $f_j'(t) = \frac{c_j}{2}(n_j^0 + \nu + t)^{-3/2} \geq 0$ and $f_j''(t) = -\frac{3c_j}{4}(n_j^0 + \nu + t)^{-5/2} \leq 0$ (well defined at $n_j^0 = 0$ because $\nu > 0$), hence is nondecreasing and concave. Since $S \mapsto \sum_{e \in S} a_{ej}$ is nonnegative and modular, its composition with the concave f_j has diminishing returns and is submodular; nonnegative weighted sums preserve monotonicity and submodularity, and $F(\varnothing) = 0$. For F_{hole}, each $S \mapsto \mathbf{1}\{\sum_{e \in S} a_{ej} > 0\}$ is a monotone submodular coverage function. The standard greedy guarantee then applies. □

Why submodularity is the right property. It formalizes diminishing returns: the tenth practice episode in a cluster helps less than the first. Greedy selection is then guaranteed within a $1 - 1/e \approx 63\%$ factor of the best allocation *of this surrogate*, so the *statistical* component of acquisition is near-optimal.

Remark 11 (What the guarantee does and does not cover). Proposition 8 concerns the frozen-round surrogate F with known deterministic increments a_{ej}; it justifies only the statistical (coverage-deficit) component $c_j/\sqrt{n_j + \nu}$ of the certificate width. It does *not* certify the full loop: stochastic feasibility, the round-to-round refit of the encoder and pair law, and the bias terms are outside its scope. Extending the guarantee to stochastic, adaptive increments would require an adaptive-submodularity analysis, which we do not claim.

K Simulator Verification

Before spending any API budget we verified every finite-sample prediction in a ground-truth simulator whose cell effects are known (checks P1–P7). *(P1)* The Theorem 1 radius $\sqrt{2C_2 t_\delta/N} + 4Bt_\delta/(3N)$ is a valid envelope of the realized sup-error at every $N \in \{100, \ldots, 2000\}$ (violation rate $\leq \delta$). *(P2)* The accepted-mean error scales as $n_{\text{eff}}^{-1/2}$ (log–log slope -0.5, Lemma 4). *(P3)* The pessimistic selector's true value is non-negative in $400/400$ resamples (Theorem 3). *(P4)* The empirical conformal per-pair envelope attains nominal coverage (Proposition 6; the top-k form is Proposition 2). *(P5)* Splitting the fiber budget as $\beta_{\gamma/k}$ is *necessary*, with k=4 the correctly split certificate covers at $0.885 \geq 1 - \gamma$ while the unsplit β_γ collapses to 0.54, exactly the $1 - \delta - k\gamma$ degradation predicted by Proposition 7. *(P6)* A coverage hole's excess loss stays flat at 0.032 as N grows $20\times$ while the matched region's loss vanishes, and the constructed two-environment pair meets the $\alpha\Delta/2$ minimax bound with equality (Theorem 2). *(P7)* Greedy acquisition attains the exhaustive optimum (ratio $1.0 \geq 1 - 1/e$, Proposition 8).

L Full Experimental Protocol

Setup. The controlled benchmark comprises 16 tool domains, each with a read-only and a destructive tool over an identified object (flight status/cancel, calendar get/delete, order get/return, file read/delete, hotel, git branch, cloud instance, playlist; deployment uses 8 *held-out* domains: email,

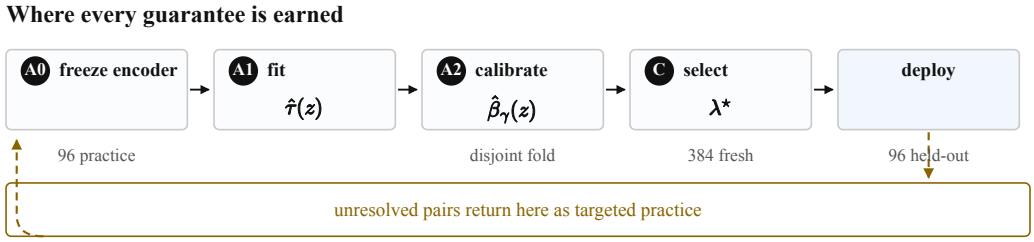


Figure 5: Each estimator is frozen before the fold that audits it, so no guarantee is earned on data used to fit it. **A0** fixes the encoder, metric and cell partition before any calibration, which is what keeps conformal exchangeability from being broken by tuning cells to outcomes; **A1** fits the transfer estimate and the shortlist map; **A2** forms the conformal envelope and per-pair radius on a disjoint fold; **fold C** selects the scale by the pessimistic rule on 384 fresh tasks; and deployment runs on 96 held-out-domain tasks. Unresolved pairs return to A1 through active coverage repair. On the confirmatory fold the pessimistic value was $\widehat{J}_C = 0.163$ with a one-sided margin of $+0.039$, selecting $\lambda^\star = 0.25$.

database record, subscription, ticket, domain registration, webhook, smart device, ad campaign). Each domain yields requests at three ambiguity levels (*explicit*, *cued*, *blind*) $\times$ two phrasing variants, giving 96 practice and 96 deployment tasks. On the primary executor the practice side is later organized as a pilot/confirmatory design for the formal selector study: the pilot (the original 96 plus a 96-task extension) freezes all estimators, and a fresh pre-specified (before any confirmatory data were collected) 384-task confirmatory fold is evaluated once. The original 96 deployment tasks are never extended or re-run; a second pre-specified 384-task fold (fold D) later adds *fresh* tasks in the same eight deployment domains, so that the frozen policy can be evaluated on the deployment law directly, and it too is evaluated once (Appendix O). Every task is run under *none* plus all eight memory types (supportive, lexical distractor, structural match, partial, stale, corrupted, conflicting, irrelevant) with $R = 3$ repetitions: 5,184 scored episodes (768 practice and 768 deployment pair units) for the primary executor, plus 768 LLM-judge calls, full-benchmark replications on two further model families (Qwen3-32B and GPT-OSS-120B, 6,912 episodes), and a frontier replication on Claude Sonnet 4.5. The primary executor is Claude Haiku 4.5. The retriever returns all $k = 8$ candidates per request (each surface-matched to the request); we rank candidates by train-fold $\widehat{\tau}$ (outcome-independent of the calibration fold), shortlists the top $K{=}2$ (map frozen on the training fold), and forms conformal intervals with the per-candidate allocation of Proposition 2: for the certified gate, envelope and rollout radius both at $\alpha/(2K)$ with $\alpha = 0.1$ (at $R{=}3$ the rollout radius $\rho_R(\alpha/(2K)) = 1.71$ dominates, which is part of why one calibration round cannot certify per-pair signs); for the average-value selector, the envelope at γ/K, $\gamma = 0.1$, scaled by $\lambda^\star$. The scale $\lambda^\star$ is selected *on practice only* by the pessimistic candidate-set rule of Proposition 1: on the primary executor with the corrected empirical-Bernstein radius of Corollary 1 on the fresh confirmatory fold C, on the other families with a task-cluster bootstrap radius at $N{=}96$ (labeled empirical), with no access to deployment outcomes. The primary executor's task-level fold assignment is: pilot (192 tasks; fits $\widehat{\tau}$, cells, shortlist map, and fixes $|C|{=}384$ before collection), fold C (384 fresh tasks; computes the selector statistic once), deployment (96 held-out-domain tasks; evaluation only), no task appears in more than one role. *Operational meaning of defer:* a deferred task is executed by the no-memory base agent and scored by that outcome, so defer and fallback are operationally identical here (both cost one base-agent episode); they differ in the *signal* they emit; defer marks the pair as an acquisition target for active repair (§5), fallback marks the candidates as certified-harmful.

Outcomes, statistics, and hygiene. The primary outcomes are the mean paired gain τ and the negative-transfer rate; harmful ground truth comes from direct repeated rollouts on a disjoint set (never from our own estimator); the resampling unit is the task (candidate memories and seeds are within-cluster); each model family is reported separately; and the conformal envelope is calibrated from pair *means*, not raw $\widetilde{\Delta}$, so rollout noise is not mistaken for fiber heterogeneity.

Compute totals. 146,261 cached episode records in all: 127,701 hosted-API rollouts and 18,560 local AWQ-served Qwen3 rollouts. Certification dominates, at 92,160 records across two separate 46,080-episode collections, the exploratory round of Appendix M and the frozen validation of Appendix T, the latter costing ≈$107 measured from its own token counts. The remainder spans three model families (Claude, Qwen, GPT-OSS, the latter two open-weight) and includes the practice pilot extension, both 384-task confirmatory folds (practice-side fold C at 3,456 episodes and held-out-domain fold D at 1,536, the latter ≈$3.47; Appendix O), the BFCL replication and a 342-rollout AppWorld probe; the 18,560 locally served episodes are the seed-pairing and size-ladder studies (Qwen3 8B/14B/32B AWQ via vLLM on a single 24 GB A10G). The per-track ledger is Table 5.

M CERTIFICATION ROUND

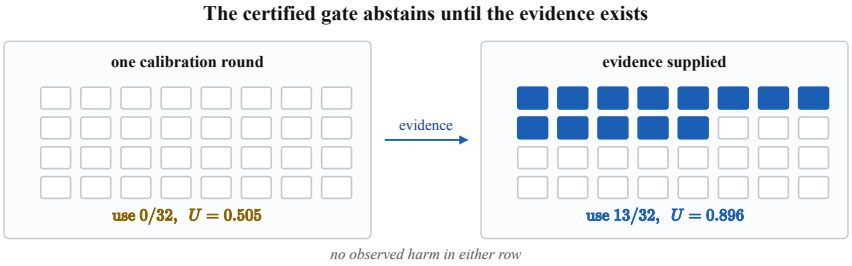


Figure 6: At the initial evidence level the certified gate ($\lambda = 1$) defers on every one of the 32 blind deployment tasks, which is the honest outcome: one calibration round leaves within-cell heterogeneity that no per-pair sign is certifiable against. Adding 96-rollout pair means, at least 39 residuals per cell, and a decision-time base-agent probe entering the probe-augmented encoder h^+ opens use on 13 of 32 tasks with no observed harm. That second row is *exploratory*: the representation was chosen after inspecting coverage on these same tasks. An independent fold of 46,080 episodes with the representation frozen in advance then certified nothing at all, deferring on all 32 tasks at U=0.496, the never-memory value. The exploratory opening does not replicate; the first and third rows are the reportable ones.

If the certified gate's deferral is an evidence deficit and not pessimism, supplying the evidence the propositions ask for should open it. We ran a dedicated certification round on the blind deployment slice (the only regime with upside) with evidence scaled to the theory's requirements: 128 calibration tasks (8 practice domains × read/write × 8 phrasing variants) and *matched rollout budgets on both sides*: each side used 96 base-agent rollout slots per arm, of which four are reserved for the pre-decision probe, leaving R_{eff}=92 matched rollouts for the calibration and evaluation pair means alike, so the calibration residuals and the deployment-side pair means share the same noise scale, as Proposition 6 requires (mixing 8-rollout calibration means with 96-rollout test means would break exchangeability). Each certifying cell has $\geq$ 39 conformal residuals as $q{=}\alpha/(2K) = 0.025$ demands, and $\rho_R(0.025; R_{\text{eff}}) = 0.309$, down from 1.71 at R=3. Two lessons emerged in sequence. *First, rollout precision alone is not enough*: with the small ρ_R but the provenance×recommended-action cell map, every $\widehat{\beta}_{0.025}$ remained ≈0.7 and the gate still deferred; the spread is genuine fiber heterogeneity (whether the *base* agent already resolves the blind request varies within cells), not noise. *Second, a decision-time probe resolves it*: we add to z a band of the no-memory probe (the mean of R_0=4 base-agent rollouts, measurable before any memory decision). *Why this is a post-hoc diagnostic, stated precisely.* Two steps must be kept apart. The *choice* of representation was made after seeing the coverage numbers above: we observed that the provenance×recommended-action map left every $\widehat{\beta}_{0.025}$ near 0.7, and we then added the probe band in response. That is a model-selection step informed by the same tasks, and it is why we call the result a nominal diagnostic rather than a certificate, and why we do not report it as validated coverage. *Conditional on* that choice, the calibration was then run consistently: the probe feature and its binning were fixed before any calibration residual for the new representation was computed, and fresh rollout noise was collected, so the interval reported is the interval the frozen construction prescribes. This does *not* restore task-level exchangeability and does not validate coverage: the same tasks informed the choice of representa-

λ	$\widehat{J}_C$	R^-	$R^\pm$	$\widehat{J}_C - R^-$	$\widehat{J}_C - R^\pm$
π_0 (defer-all)	0.000	0.000	0.000	0.000	0.000
0.25	0.163	0.124	0.137	**+0.039**	**+0.027**
0.5	0.163	0.124	0.137	+0.039	+0.027
0.75	0.163	0.124	0.137	+0.039	+0.027
1.0	0.000	0.065	0.073	−0.065	−0.073

Table 6: Confirmatory-fold pessimistic scores. Both the safety selector ($\arg\max \widehat{J}_C - R^-$) and the oracle selector ($\arg\max \widehat{J}_C - R^\pm$) return $\lambda^\star$=0.25 (ties at $0.25/0.5/0.75$ broken toward the smallest scale by the pre-fixed rule); λ=1 defers everywhere so its accepted-region increment is 0.

tion, so re-drawing rollouts on them removes rollout-level reuse but not the selection effect. The two statements are compatible in sequence, and neither rescues the other. Establishing the formal claim requires the representation and its binning to be fixed in advance and evaluated on a fold not used to discover the problem. That was not available for this exploratory result; we report exactly such an independent validation, and its negative outcome, in Appendix T. Protocol hygiene: the probe feature and its binning (threshold 0.5, R_0=4) were fixed *before* this round's calibration data were collected, and probe reps are excluded from evaluation and residual means on *both* the calibration and deployment sides. The nominal intervals then separate: the two use-triggering cells (·, rec-read, base-low) have $(\widehat{\tau}, \widehat{\beta}_{0.025}) = (0.950, 0.592)$ and $(0.998, 0.640)$, and with $\rho_R(0.025; R_{\text{eff}}{=}92) = 0.309$ the smallest nominal lower bound is $\min_{\text{use-triggering cells}} \text{LCB} = +0.049 > 0$ at λ=1: positive but thin, which is the honest picture; base-high cells retain $\tau \approx 0$ intervals and do not trigger use. Deployed at λ=1, α=0.1, this *exploratory* probe-augmented gate uses memory on 13 of 32 blind tasks and defers on 19 (an independent frozen-representation fold later produced $0/32$; Appendix T): mean utility $0.505 \to 0.896$ (oracle 1.000) at observed 0% negative transfer ($0/13$ harmful uses, exact 95% upper bound 20.6%). The probe is not free, so we price it: it costs 4 extra base episodes per decision, and a *probe-only* control that spends the same budget without any memory decision (majority vote over the 4 probe rollouts) reaches only 0.531 versus never-memory; the gain comes from the probe-augmented memory decision, not from the extra executions. Deferral was an evidence deficit, and what opened these decisions is a statement about how much evidence the deployment demands: rollouts, calibration pairs, *and* representation resolution. Because the last of those was chosen on this data, the 13 uses are nominal rather than certified; the residual 19 defers are the next acquisition targets, closing the loop with §5.

N Confirmatory Fold Protocol

At the original N=96 practice tasks, no theorem-backed radius admits a non-trivial formal selection: the Hoeffding radius ($\sqrt{2\log(2|\Pi|/\delta)/N} = 0.332$, δ=0.05, $|\Pi|$=5) and the corrected empirical-Bernstein radius of Corollary 1 (0.35 at the observed $\widehat{V}_\pi \approx 0.13$; the range-2 correction matters: the uncorrected $[0,1]$ constant would give 0.25) both exceed the observed policy means (0.15–0.17), so the formal selectors retain π_0 and only a task-cluster bootstrap, an *empirical* instantiation: picks $\lambda^\star$=0.25. To obtain a formal selection without adaptive-sample-size or reuse bias, we followed a pilot/confirmatory protocol: *all* previously collected practice (192 tasks) was demoted to a pilot, used only to freeze $\widehat{\tau}$, the cells, and the shortlist map, and to fix the confirmatory fold size from the corrected formula (pilot $\widehat{J} \approx 0.17$, $\widehat{V} \approx 0.14 \Rightarrow$ margin positive from $N \approx 288$; we fixed N=384, a factorial-exact size, before collecting any confirmatory data). The confirmatory fold C (384 fresh tasks, new surface templates and blind phrasings, disjoint from every prior task id) was then evaluated *once*, with no interim looks. Result: $\widehat{J}_C = 0.163$, $\widehat{V}_C = 0.129$, corrected one-sided radius 0.124 and two-sided 0.137. Both formal selectors then pick $\lambda^\star$=0.25: the pessimistic score $\widehat{J}_C - R^-$ (resp. $\widehat{J}_C - R^\pm$) is positive at $\lambda \in \{0.25, 0.5, 0.75\}$ and maximized at λ=0.25, while λ=1 collapses to the defer-everywhere policy (Table 6); the safety selector $\widehat{\pi}^-$ and the oracle selector $\widehat{\pi}^\pm$ therefore return the same λ, agreeing with the bootstrap. Deployed on the untouched 96 held-out-domain tasks, the formally selected policy reproduces the headline result exactly: mean utility 1.000, observed 0% negative transfer, use on all 32 blind tasks.

The *practice-side selection* is thus theorem-backed on the primary executor (Proposition 1 + Corollary 1 on fold C); its deployment performance is a held-out empirical evaluation, with transfer subject to the declared sensitivity term $\epsilon_{\text{rep}}^{C}$. For the other two families we report the bootstrap selection at $N{=}96$ and label it empirical.

Sampling law of the confirmatory fold. The confirmatory fold is a *pre-fixed balanced grid*: strata are (domain × intent × ambiguity), and within each stratum a fixed number of phrasing variants is generated independently, giving 384 independent task outcomes whose per-task expectations are fixed by the design. We apply Corollary 1 with the confirmatory target defined as the average of these pre-specified per-task expectations, $J_C(\pi) = \frac{1}{N}\sum_i \mathbb{E}[X_i^\pi]$. Because the outcomes are independent (not identically distributed) and the grid is fixed before collection, the independent empirical-Bernstein inequality (Maurer & Pontil, 2009, Thm. 11) applies directly. This is deliberately a *fixed-design* reading; the guarantee is about the balanced-population mean of the pinned-down grid, and *not* a claim of unconditional i.i.d. sampling from a natural deployment law Q_C.

What was frozen before fold C. For the finite-class bound to apply, everything the selector depends on must be fixed before the confirmatory outcomes are seen. Before collecting fold C we froze: the retriever, the encoder h, the metric and cell partition, the estimator $\widehat{\tau}$, the conformal envelope $\widehat{\beta}_\gamma$, the rollout radius, the score $s(z)$, the shortlist map, the scale grid Λ_{AVG}, the tie-breaking rule, the task generator, and the stratum weights. None of these was adjusted after observing confirmatory outcomes; had any been, the finite-class bound would not apply as stated.

O HELD-OUT-DOMAIN CONFIRMATORY FOLD (FOLD D)

Fold C is built from the eight *practice* domains, so its bound is a statement about the practice side and carrying it to deployment costs the declared sensitivity term $\epsilon_{\text{rep}}^{C,\text{grid}}$. This appendix removes that term by measuring the frozen policy on the held-out deployment domains themselves. Like fold C, fold D is a *fixed-design balanced grid*, not an i.i.d. sample from a natural deployment law, so the target is $J_D(\pi) = \frac{1}{N}\sum_i \mathbb{E}[X_i^\pi]$, the average of the per-task expectations pinned down by that grid, exactly as J_C is defined for fold C (Appendix N). Every bound below is a statement about J_D and about that grid; it is *not* a statement about an unconditional deployment distribution.

Construction. Fold D applies fold C's surface-variation scheme to the eight *held-out deployment* domains: 8 domains × 2 intents × 3 ambiguity levels × 8 fresh phrasings = 384 tasks. The eight explicit and eight cued templates are reused verbatim from fold C so that the only difference between the two folds is the domain set; the eight blind phrasings are new and disjoint from those of every prior track. Task identifiers carry an `x_` prefix and disjointness from every previously collected identifier is asserted programmatically before collection. The 96 deployment tasks of Table 2 are a separate held-out set and remain untouched.

What was frozen, and what is therefore claimed. Nothing was refitted for fold D. The encoder h, the cell partition, $\widehat{\tau}$, the conformal envelope $\widehat{\beta}_\gamma$, the shortlist map, the candidate generator, the scorer, the executor family (Haiku 4.5, temperature 0.8, $R{=}3$) and the scale $\lambda{=}0.25$ were all carried over unchanged, λ having been selected by fold C and *not* reselected here. Because h is computed from text alone, the policy's decision on a new request needs no probe rollouts, so the per-task decisions were written to a manifest before any episode was collected. One policy is evaluated, hence $|\Pi|{=}1$ and the union bound over the λ grid disappears: the one-sided radius uses $\ln(2/\delta)$ rather than $\ln(2|\Pi|/\delta)$, which is the entire statistical benefit of freezing. The linear term keeps the range-2 constant 14 of Corollary 1. The analysis formula, $\delta{=}0.05$, $N{=}384$ and the decision threshold were all pre-specified. Substituting the *pilot* variance 0.1294 for power planning gave an anticipated radius of 0.0948, which is what made $N{=}384$ the planned size; the same pre-specified formula applied to fold D's *observed* variance 0.0976 yields the radius actually reported, 0.088. The pilot number is a planning quantity and plays no part in the reported bound.

Result. The primary endpoint was fixed in advance as $\text{LCB}_{0.95}[J_D(\pi)] > 0$, evaluated once, with no interim looks and no reanalysis. Abstention and fallback require no additional *memory-arm* episodes, since the policy then plays no-memory and the paired gain is defined as exactly zero;

the no-memory episodes are still collected on all 384 tasks so that absolute utility can be reported. This is why the ledger shows 1,536 episodes rather than only the used-arm pairs: 384 tasks $\times$ 3 repetitions on the no-memory arm, plus 128×3 on the memory arm the policy plays.

	fold C (practice domains)	**fold D (deployment domains)**
N	384	384
policies evaluated $\lvert\Pi\rvert$	5	**1**
$\widehat{J}$	$+0.163$	$+0.120$
$\widehat{V}$	0.1294	0.0976
one-sided radius	0.124	0.088
$\mathrm{LCB}_{0.95}$	$+0.039$	$+0.032$
two-sided companion	$+0.027$	$+0.019$
transport term required	$\epsilon_{\mathrm{rep}}^{C,\mathrm{grid}}$	**none**
uses	—	$128/384$
harmful uses	—	$0/128$ (exact 95% upper 0.023)
utility, policy vs never memory	—	0.948 vs 0.828
per-domain mean gain	—	positive in $8/8$, $+0.104$ to $+0.139$

Table 7: The average-value claim opens on both grids. Fold D is the stronger statement despite the smaller point estimate, because it needs no practice-to-deployment transport term and evaluates a single frozen policy. Both targets are fixed-design grid means, J_C and J_D. Cost: 1,536 episodes, $\approx$\$3.47.

Reading, including what this does not establish. The bound opens: $\mathrm{LCB}_{0.95}[J_D(\pi_{\text{TRACE-AVG}})] = +0.032$ on the held-out-domain grid, with no sensitivity parameter to declare. The point estimate is the smaller of the two ($+0.120$ against $+0.163$). That 0.043 is the *selected-policy grid gap*, observed rather than absorbed into an unknown term, and it is what $\epsilon_{\mathrm{rep}}^{C,\mathrm{grid}}$ was previously standing in for at this policy. It is *not* an estimate of the supremum transport error over policies or distributions, which fold D does not bound; what it does show is that the sensitivity term was carrying real uncertainty rather than being a formality. The use rate is $128/384$, the same one-third the policy takes on the original deployment tasks, and no use is harmful. Rather than a domain-clustered interval we report the raw spread: the mean gain is positive in all $8/8$ held-out domains, from $+0.104$ to $+0.139$. A clustered 95% interval on eight such domains comes out implausibly narrow because they are near-replicates of one another by construction, so an interval would overstate the precision of a claim about domains at large; the small domain base remains a limitation (Appendix S). Two further caveats bound the reading. The claim is about J_D on the grid *we specified*, not about an unconditional deployment law or about arbitrary distribution shift; a genuinely different deployment law would need its own fold. And the deploy domains supply a single cue sentence per intent, so fold D's surface variation comes from the templates alone and is narrower than fold C's.

P Cross-Family Analysis

Cross-family replication (Table 8). We repeat the full benchmark on two further model *families*: Qwen3-32B and GPT-OSS-120B (6,912 additional episodes), and the blind subset on Claude Sonnet 4.5. The dominant sign pattern replicates on every executor: supportive and structural-match memories help equally, lexical distractors and corrupted memories hurt, irrelevant memory is inert. Recoverable-defect memories are the exception that proves the per-pair point: *partial* memories are positive on the Anthropic and Qwen executors but essentially zero on GPT-OSS. The *magnitudes* are diagnostic of the executor: the distractor penalty tracks how faithfully the model follows memory (-0.25 Haiku, -0.36 GPT-OSS, -0.66 Qwen3; Sonnet on blind tasks: supportive $+0.516$ vs. distractor -0.484), the corrupted penalty is severe on all three (-0.42 to -0.60), and GPT-OSS is uniquely brittle to *partial* memories. Under *explicit* or *cued* phrasing every memory type yields $\tau \approx 0$ on every model: memory only matters when the request underdetermines the action.

Within the evaluated ladder, the observed penalty grows with size. On a locally served Qwen3 size ladder (8B/14B/32B, all AWQ-quantized; 10,368 episodes) the dominant sign pattern is identi-

Memory type	Haiku 4.5	Qwen3-32B	GPT-OSS-120B
supportive	$+0.17$	$+0.18$	$+0.15$
structural match	$+0.17$	$+0.18$	$+0.15$
stale	$+0.17$	$+0.19$	$+0.15$
partial	$+0.16$	$+0.12$	-0.03
conflicting	$+0.14$	$+0.17$	$+0.15$
irrelevant	-0.01	-0.02	-0.03
lexical distractor	$\mathbf{-0.25}$	$\mathbf{-0.66}$	$\mathbf{-0.36}$
corrupted	$\mathbf{-0.42}$	$\mathbf{-0.51}$	$\mathbf{-0.60}$

Table 8: τ by memory type across three model families (96 practice tasks each, cluster-bootstrap CIs in the experiment logs). The dominant sign pattern replicates, while recoverable defects such as partial memories remain executor-dependent.

Executor	γ=.05	γ=.1	γ=.2	γ=.3
Haiku 4.5	1.000	0.978	0.893	0.841
Qwen3-32B	0.975	0.973	0.888	0.820
GPT-OSS-120B	0.992	0.966	0.906	0.858
nominal $1-\gamma$	0.950	0.900	0.800	0.700

Table 9: Live-data conformal coverage (leave-one-task-out per structural cell) vs. nominal. Every aggregate entry meets or exceeds its target: empirically consistent with Proposition 6 at the population level; the adversarial sub-population sweep in the text locates the one slice where the transport assumption fails.

cal at every size, but the harmful-memory magnitudes are *monotone in scale*: the lexical-distractor penalty grows $-0.35 \to -0.49 \to -0.67$ from 8B to 32B. Larger models are *better* at following retrieved instructions, which makes a structurally wrong memory *worse*, so scaling the executor is not a mitigation and makes a measured per-pair gate more, not less, necessary. Recoverable-defect handling improves with scale in the same data (partial: $+0.15 \to +0.17 \to +0.18$; conflicting: $+0.15 \to +0.04 \to +0.17$ with the mid-size dip an executor quirk).

Live-data diagnostics are favorable, with one diagnosed transport failure (Table 9). *(i) Conformal validity* (Corollary 4(b), Proposition 6): leave-one-task-out empirical coverage of the cell-conformal interval meets or exceeds its nominal $1-\gamma$ at every $\gamma \in \{0.05, 0.1, 0.2, 0.3\}$ on every model. *(ii) Representation invariance* (ϵ_{rep}, Assumption 2): the practice-estimated cell means predict the *held-out-domain* deployment cell means with correlation $+0.996/+0.998/+0.986$ (Haiku/Qwen3/GPT-OSS) and sign agreement $8/9$, $9/9$, $8/9$. *(iii) Cross-executor stress test*: a gate calibrated *entirely on Haiku practice* still helps when deployed on the other families: Qwen3 $0.784 \to 0.958$ and GPT-OSS $0.812 \to 1.000$ mean utility at 0% negative transfer: showing the structural cells transfer even across model families. This residual utility gap is an empirical cross-executor transport diagnostic, not a valid upper bound on ϵ_{rep}. *(iv) Residual transport, stressed until it breaks*: calibrating on practice and covering *deployment* pairs meets nominal at every γ on every family (e.g. $0.988/0.971/0.951$ at γ=0.1). We then adversarially restrict deployment to sub-populations *inside* the cells (each ambiguity level, each intent, each of the eight domains: 14 slices per family): 41 of 42 slice–family combinations still meet nominal; the single violation is GPT-OSS on the blind-only slice ($0.879 < 0.90$), a measured instance of exactly the failure mode the residual-transport caveat names. *(v) Rank-conditional calibration was necessary in this audit and restored aggregate nominal coverage*: on the deployed K=2 shortlist the residual law *does* differ by shortlist rank (two-sample KS, $p < 10^{-3}$ on all three families), and the pooled envelope under-covers on GPT-OSS (0.880 vs. nominal 0.90); calibrating the envelope rank-conditionally, the remedy §4.3 prescribes: restored nominal coverage on all three evaluated families ($1.000/0.969/0.974$).

Per-family deployment policies (Table 10). With each model's own practice-calibrated $\lambda^\star$, the average-value selector's action pattern is *identical* across families: use on the 32 blind tasks, defer on the 64 ceiling tasks, and shows observed 0% negative transfer everywhere: Haiku 1.000 (oracle-matching), GPT-OSS 1.000 (with the more conservative $\lambda^\star$=0.75 its calibration selects), Qwen3

Executor	$\lambda^\star$	never	always(sim)	supervised	CERT	AVG	oracle
Haiku 4.5	0.25	0.826	0.979 (2.1%)	0.826	0.826	**1.000** (0%)	1.000
Qwen3-32B	0.25	0.784	0.982 (1.0%)	0.784	0.784	**0.958** (0%)	1.000
GPT-OSS-120B	0.75	0.812	0.990 (0%)	0.812	0.812	**1.000** (0%)	1.000

Table 10: Deployment policy comparison per model family (96 held-out-domain tasks each; parentheses give observed negative-transfer rate). CERT = the certified gate (λ=1; defers everywhere after one calibration round), AVG = the average-value selector ($\lambda^\star$ selected per family on practice only).

Axis (controlled harmful regime)	always-sim	CMI	the average-value selector
Utility U / observed harm	0.979/2.1%	1.000/**0**%	**1.000/0**%
Extra decision probes / request	0	51	**0**
Scorer calls / request	0	51	**0**
Upfront (empirical / confirm.)	0	0	2,592 / 8,640
Break-even $N^\star$ vs CMI	–	–	51 / 170
Guarantee of compared policy	none	none	pract. avg.

Table 11: Controlled head-to-head with an adaptation of the closest concurrent method (CMI) on the controlled *harmful* regime. Both safe methods reach oracle utility at 0% harm, so the distinction is systemic rather than a utility win; the U=1.000 row is the average-value regime, *not* pair-certified, and CMI is given an oracle request-time scorer it would lack in deployment. Protocol, cost derivation and the adapted RSCB-MC / ToolChain-CRC stress tests are in Appendices Q–Q.1.

0.958 vs. oracle 1.000 (the residual gap reflects Qwen3's stronger memory-following). The certified gate (λ=1) defers everywhere on all three families after one calibration round, equalling the never-memory baseline (0.826/0.784/0.812).

Q Head-to-Head Baselines

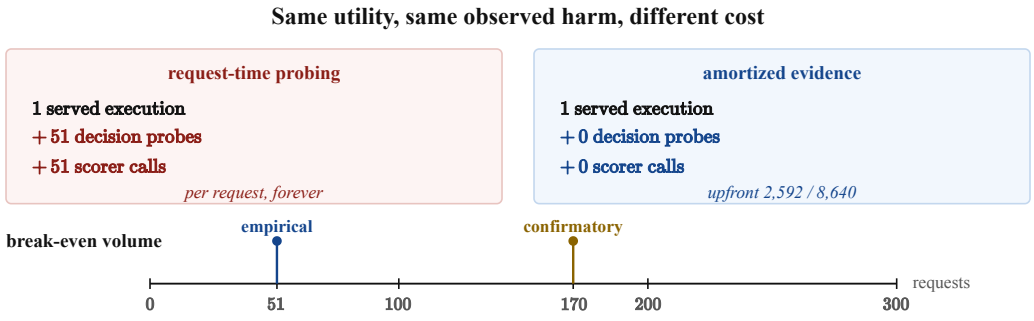


Figure 7: A controlled head-to-head against an adaptation of the closest concurrent method on the controlled harmful regime, with $k = 8$ candidates and $R = 3$ rollouts per probe. Both reach utility 1.000 at 0% observed harm, so the comparison is not about accuracy: the prior method pays 51 extra counterfactual probes and 51 outcome-scorer calls at *every* request, and needs a request-time scorer that ordinary deployment does not have, whereas our evidence is collected once before deployment. The break-even deployment volume is 51 requests against the empirical budget and 170 against the confirmatory one, after which we issue strictly fewer model calls.

We compare against *adapted proxy baselines inspired by RSCB-MC (Iscan, 2026) and ToolChain-CRC (Opoku & Banahene, 2026)*. Both are decision *rules* over a retrieved memory, so we apply our implementations to the *same* measured pair values and the *same* practice/deploy split, holding executor and benchmark fixed (a controlled contrast of rules). We stress that these are *simplified adaptations* of the methods described in these concurrent preprints, not faithful reproductions. Our RSCB-MC proxy is a risk-sensitive contextual bandit that learns a value model and a risk model

	Controlled		BFCL V4	
Decision rule	mean U	neg.	mean U	neg.
never-memory	0.826	0.0%	0.968	0.0%
always (similarity)	0.979	2.1%	0.983	3.0%
RSCB-MC (adapted proxy)	0.969	2.1%	0.960	3.0%
ToolChain-CRC (adapted proxy)	0.979	2.1%	0.983	3.0%
Avg. selector	**1.000**	**0.0%**	0.968	**0.0%**
Certified gate	0.826	0.0%	0.968	0.0%
oracle	1.000	0.0%	0.995	0.0%

Table 12: Adapted proxy baselines on identical measured pair values and splits (observed negative-transfer rates). Among the rows shown, the relevance/risk-scoring rules track always-use in both harm and utility, while the coverage-certified/pessimistic rules keep observed harm at 0%. The proxies simplify the originals, so the differences may partly reflect the adaptations.

from 4 generic relevance/metadata features (similarity, provenance-surface, memory length, retrieval rank: pointedly *not* the structural transfer estimate) and accepts when $\widehat{r} - \kappa\widehat{s} > 0$ with κ tuned on practice; the original RSCB-MC operates over a *multi-action* space with a richer 16-feature representation, whereas our proxy uses these 4 features and a binary accept/reject threshold. Our ToolChain-CRC proxy calibrates a *single* conformal threshold on a similarity-derived danger score so the practice harm rate is $\leq \alpha$=0.05; the original ToolChain-CRC operates at the level of *step-level trajectory risk* with an anytime alarm, whereas our proxy applies one conformal threshold to a single scalar danger score at the episode level.

These surface-score adaptations provide a targeted stress test of whether relevance- or risk-ranking *alone* can detect the structural sign reversal: scoring relevance/risk rather than incremental transfer, *these particular adapted proxies fail to suppress negative transfer on our twin-trap data*: on the controlled benchmark both pay 2.1% harm (RSCB-MC 0.969, CRC 0.979), matching always-use-similarity, while the average-value selector reaches 1.000 at 0% harm; on BFCL both pay 3.0% harm (RSCB-MC 0.960, CRC 0.983) where the pessimistically selected average-value rule abstains at 0% (Table 12). The CRC proxy controls harm only to the level its similarity score can *rank*, which in a twin trap it cannot; the RSCB-MC proxy's risk model, built on four surface/metadata features, does not resolve the structural sign flip. Because these are simplified adaptations, the observed gaps may partly reflect the adaptations rather than the methods as described in full; the point is narrow, the four surface/metadata features used in these adaptations do not resolve the structural sign flip in our benchmark, not that the original paradigms cannot.

Q.1 Adapted CMI on the controlled harmful regime

The organic BFCL-Multiple comparison (§Z) placed adapted CMI in a ceiling regime where the distractor harms $0/39$ tasks, so no method has to resolve a real safety decision. To close this, we run the *same* adapted CMI implementation on the primary controlled benchmark, where corrupted and lexical-distractor memories are genuinely harmful ($\tau = -0.42$ and -0.25; Table 1). We reuse the identical 96 deployment tasks, the eight memory types, the exact ground-truth scorer, and the *cached* no-memory and with-memory rollouts from the main controlled run; the only new rollouts are the per-candidate *perturbed*-memory probes CMI requires (deploy $\times 8$ kinds $\times R$=3; the perturbation redacts the concrete tool/argument the memory's playbook commits to, matching the Track-S convention).

Scorer disclosure. CMI needs a request-time outcome score for each no-memory, with-memory, and perturbed-memory probe; we provide it the benchmark's exact ground-truth evaluator as an oracle request-time scorer (the same evaluator for all three probe types). This *favors* CMI and is unavailable in ordinary deployment. We use no request-time scorer.

CMI selection rule (exact protocol). Scores are aggregated as the mean over the R=3 paired rollouts of each probe type. A candidate m is *eligible* if $\mathrm{util}(m) = \bar{u}_{\mathrm{with}}(m) - \bar{u}_0 > 0$ and $\mathrm{stab}(m) = \bar{u}_{\mathrm{with}}(m) - \bar{u}_{\mathrm{pert}}(m) \geq 0$; CMI selects the eligible candidate with maximum util, ties broken by larger stab then by candidate index, and falls back to no memory if none is eligible. The final deployed episode is a fresh execution with the selected memory (the probe rollouts are decision-

Cost / guarantee axis (k=8, R=3)	always-sim	CMI	the average-value selector
Deployment utility / observed harm	0.979/2.1%	1.000/0%	1.000/0%
Harmful uses (exact-binom. 95% UB)	–	0/19 (14.6%)	0/32 (8.9%)
Final task execution / request	1	1	1
Extra decision probes / request	0	51	0
Total model calls / request	1	52	1
Outcome-scorer calls / request	0	51	0
Upfront calls: empirical (N=96)	0	0	2,592
Upfront calls: confirmatory	–	–	8,640
Break-even $N^\star$ vs CMI (empir./conf.)	–	–	51 / 170
Guarantee of *compared* policy	none	none	practice-side avg.
Separate pair-certificate available	no	no	yes (the certified gate)

Table 13: Adapted CMI vs. the average-value selector on the controlled *harmful* regime, where corrupted/distractor memories genuinely hurt. Both reach oracle utility at 0% observed harm (CMI given an oracle request-time scorer it would not have in deployment). Every method runs one final execution; CMI adds 51 decision probes and a scorer call each, ours none. We amortize evidence upfront: 2,592 rollouts for the empirical selector, 8,640 (pilot + fold C) for the confirmatory one: breaking even at $N^\star \approx 51/170$ requests. The compared policy is the average-value selector (average-value guarantee only); The certified gate is a separate regime not applied to this row.

time experiments, not reused as the served response), so CMI's per-request model calls are the $2k{+}1$ probe executions ($\times R$) *plus* the one final execution; we report the decision probes separately from that shared final execution.

Result. With this oracle scorer, adapted CMI selects a memory on $19/96$ tasks and reaches mean utility 1.000 at 0% observed harm, with $0/19$ harmful uses ($0/96$ requests): i.e., in a regime with truly harmful memories CMI's request-time counterfactual+stability test *does* avoid them. The average-value selector reaches the same 1.000 at 0% (use $32/96$, $0/32$ harmful uses, $0/96$ requests), and always-similarity pays 2.1% harm. Applying the same disclosure standard to both, a $0/n$ harmful-use count is not strong evidence of a zero population harm probability: the 95% one-sided exact-binomial upper bound is 14.6% per use for CMI (n=19) and 8.9% for the average-value selector (n=32). So the two closest safe methods are utility- and harm-equivalent here; they differ in *where* the cost is paid and in *what is guaranteed* (Table 13).

Cost accounting. Both methods run one final task execution per request. The difference is the decision-time probes: CMI runs (k with-memory $+\,k$ perturbed) $\times$ R probes and a no-memory rollout ($\times R$): 51 probe executions, *all* of them extra beyond the one final execution that both methods run, each requiring an oracle outcome-scorer call, whereas we run 0 extra probes and need no request-time scorer. We instead pay for evidence once upfront. We report this for both instantiations, since they differ in formal status: the bootstrap-selected *empirical* (N=96) selector costs $96 \times 9 \times 3 = 2{,}592$ rollouts and carries no nontrivial formal selector, while *confirmatory* TRACE-AVG additionally consumes the pilot ($192 \times 9 \times 3 = 5{,}184$) and the pre-specified 384-task fold C ($384 \times 3 \times 3 = 3{,}456$; Appendix N), i.e. 8,640 rollouts in total. Against CMI's 51 extra calls/request the amortized cost crosses over at a break-even volume of $N^\star = 2{,}592/51 \approx 51$ requests (empirical) or $8{,}640/51 \approx 170$ (confirmatory); beyond $N^\star$ we issue strictly fewer model calls and need no request-time scorer at all. The compared average-value selector policy carries the practice-side average-value guarantee (Prop. 1); the broader framework separately provides the certified gate as a more conservative per-pair certificate (Prop. 2), but the U=1.000 row above is *not* itself pair-certified. CMI attaches no coverage-dependent guarantee.

R APPWORLD PROBE

We ran a multi-step probe on AppWorld (Trivedi et al., 2024) (57 dev tasks, 28 practice / 29 deploy, a ReAct (Yao et al., 2023) code agent with injected workflow playbooks, paired on reset worlds, scored by AppWorld's graded unit tests; 342 Claude Sonnet rollouts) to reach the upside regime. Two findings, one positive and one a limit. *Positive*: with a strong executor the estimand is well-defined and non-trivial: per-task scores span 0.1–1.0 (mean 0.505, no floor), the no-memory base-

line is graded (0.447, $53/57$ non-zero), and a supportive workflow playbook helps on deployment ($0.462 \rightarrow 0.507$ toward an oracle of 0.634). *Limit*: the clean *twin trap* does not reproduce here. AppWorld tasks are genuinely multi-app, so a "sibling-app" distractor still points at an app that is often in play, and its measured transfer is *positive* on average ($\tau = +0.11$, $[+0.01, +0.22]$), any workflow hint helps a low baseline more than a wrong-app hint hurts. The structural cell that cleanly separated harmful memories in the single-call settings therefore has no clean negative class on AppWorld, and the certified gate cannot demonstrate harm suppression where harm is not reliably inducible. This marks the boundary of the controlled construction: the sign-and-certify machinery needs a setting where a structurally wrong memory is *reliably* harmful, which our controlled single-tool and injected BFCL-Multiple constructions provide; the organic BFCL probes (ceilings) and free-form multi-app coding do not. Constructing organic multi-step twin traps, MCP-Universe (Luo et al., 2025), and the PREPING composition (Choi et al., 2026) remain future work.

S Extended Limitations

Decision unit. The decision unit is an episode-level intervention rather than a sequential policy, and the controlled benchmark resolves in a single tool call, so nothing here speaks to memories that must be revised mid-trajectory.

What the guarantees do and do not cover. The *theorem-backed* average-value statement is practice-side and fixed-design: the confirmatory margin is $+0.039$, so the implied deployment bound is $0.039 - \epsilon_{\text{rep}}^{C,\text{grid}}$ and is positive only if that declared sensitivity term is below 0.039. We do not estimate that term, and cell-mean agreement between practice and deployment is supporting evidence that effects transport rather than a bound on the gap. What we do instead is measure the held-out domains directly: a frozen policy on a fixed-design grid of 384 tasks over them gives $\text{LCB}_{0.95}[J_D] = +0.032$ with no transport term (Appendix O). The 0.043 gap between the two grids is an observed gap at the selected policy, not an estimate of the supremum transport error. That bound is about the grid mean J_D we specified, not an unconditional deployment law, and it does not extend to arbitrary shift. The eight deployment domains are near-replicates by construction, so we report the per-domain spread (positive in $8/8$, $+0.104$ to $+0.139$) rather than a clustered interval, which would overstate precision about domains at large. Pair-level certification is marginal over a random deployed shortlist, rests on a residual-transport assumption we audit rather than prove, and one adversarial cross-family slice violates it. Most importantly, an independent frozen-representation fold certifies nothing at α=0.1 (Appendix T), so the per-pair regime should be read as a specification of what evidence would be required, not as a working component.

Statistical unit. The 96 tasks are phrasing and ambiguity variants over only 8 domains, so a task-clustered bootstrap treats surface variants as independent evidence. Domain-clustered resampling, leave-one-domain-out signs and per-held-out-domain policy behaviour all leave the conclusions unchanged (Appendix V), but 8 domains remains a small base for any claim about domains at large.

Where the harm comes from. The candidate sets on all three benchmarks are authored or injected, which makes the mechanism visible and lets us check the estimand against ground truth, but also means the harmful class is present by construction rather than because retrieval produced it. Our organic settings are consistent with this: three found the nominally mismatched candidate inert or helpful, and a preregistered fourth (Appendix X) produced two isolated negative paired outcomes out of ten, including one large observed degradation, but failed its prespecified reliability gate and stayed slightly positive on average. How often high-leverage negative transfer arises in naturally accumulated experience is therefore still open. Scaling the same wrong-app design would not settle it, because it does not isolate a semantically opposed action and its two R=1 negative outcomes did not survive independent replication. Appendix Y attempts a same-app directional comparison, but its truncated workflow prefixes and coarse ontology do not form exact action twins. A decisive test would preserve the complete action-bearing procedure and freeze semantically matched opposing actions before collection.

T INDEPENDENT VALIDATION OF THE PROBE-AUGMENTED CERTIFICATE

The $13/32$ opening of Appendix M was obtained by refining the representation after inspecting coverage on the same tasks, so it is exploratory and cannot be validated by that data. This appendix reports the test that settles it: freeze everything, collect a fresh fold, look once.

What was frozen before any call was issued. The encoder $h^+ = (\text{provenance}, \text{rec-action}, \mathbf{1}\{\text{probe} \geq 0.5\})$; the probe budget $R_0{=}4$ drawn from the none-arm's first four repetitions on both sides; the evaluation reps $[R_0, 96)$, giving $R_{\text{eff}}{=}92$ matched rollouts; the level $q = \alpha/(2K) = 0.025$; the scale $\lambda{=}1$; and the train/calibration split fraction. None is a free parameter of this run. Eight new blind phrasings give task ids disjoint from every prior track.

Result: the certificate does not open. Calibration used 128 practice-domain tasks (256 pair units), deployment 32 fresh held-out-domain blind tasks, 46,080 episodes in total. Raw occupancy after the band splits the four schema cells is $22/42/49/15$ units. The binding fact is what survives the frozen train/calibration split: no deployment-relevant calibration cell retained the 39 residuals that $q{=}0.025$ demands, so the radius was infinite wherever a decision actually had to be taken, the gate deferred on all 32 tasks, and utility was 0.496, identical to never-memory. A cell needs roughly 46 raw units to retain 39 residuals after any split we tested.

Not an artefact of the split. The exploratory run reached exactly 39 residuals in two cells, so the split fraction is the natural suspect. It is not (Table 14).

train fraction	cells with $\widehat{\beta} < \infty$	uses
0.10	$2/8$	$0/32$
0.15	$2/8$	$0/32$
0.20	$2/8$	$0/32$
0.25–0.40	$0/8$	$0/32$

Table 14: Validation fold, sensitivity to the train/calibration split. At best two of eight cells obtain a finite radius, never the cells the deployment tasks occupy, so no split yields a single certified use.

A post-hoc $K{=}1$ diagnostic: multiplicity was only part of it. The shortlist allocation $q = \alpha/(2K)$ ties the residual requirement to the shortlist size, so $K{=}1$ halves the multiplicity correction and lowers the requirement from 39 to 19. Recomputing on the *same* fold is a diagnostic, not an independent validation, and we label it as such: K was not frozen in advance, and any use it produced would inherit the selection problem this appendix exists to avoid. It is reported because it isolates the cause. Table 15 shows multiplicity is no longer the main bottleneck in the cells that matter: six of eight now obtain a finite radius rather than two, though two remain unbounded at 13 units. The lower bounds nonetheless still sit just below zero, at -0.022 to -0.033 in the three cells the deployment tasks occupy, because $\widehat{\beta} \approx 0.5$–$0.8$ nearly cancels $\widehat{\tau} \approx 0.75$–$0.92$. The binding constraint is therefore genuine within-cell heterogeneity, not calibration count, and no budget increase addresses it directly: $\widehat{\beta}$ is an order statistic of that spread, and $\rho_R{=}0.283$ is already near its floor at $R_{\text{eff}}{=}92$.

What the fold does buy: the price of confidence, measured. Varying only the declared error budget traces the evidence-versus-claim curve on held-out data (Table 16). At the declared level the certificate is empty; it becomes non-vacuous only once the budget is relaxed to $\alpha \geq 0.3$. We report the curve rather than selecting a level, since selecting α after seeing $\alpha{=}0.1$ fail would repeat exactly the error this appendix corrects. Read whole, the curve measures the paper's thesis: what may be claimed is set by how much evidence has been paid for.

What this licenses. That pair-level certification at $\alpha{=}0.1$ is out of reach at this budget, and that the gate fails safe when it is: every deferral returns the never-memory arm and observed harm is zero throughout. It does *not* license the claim that more data would open it. More calibration units would make $\widehat{\beta}$ finite, but a finite radius is not a positive lower bound, and the α curve shows the margin is thin where it exists.

cell (prov, rec-act, band)	n_{cal}	$\widehat{\tau}$	$\widehat{\beta}$	LCB
(cross, read, 0)	19	$+0.750$	0.500	-0.033
(same, read, 0)	19	$+0.920$	0.670	-0.033
(same, write, 0)	42	$+0.907$	0.646	-0.022
(cross, write, 0)	42	$+0.843$	0.821	-0.261
(same, read, 1)	36	$+0.205$	0.589	-0.667
(cross, read, 1)	36	$+0.029$	0.764	-1.019
(cross, write, 1)	13	$+0.201$	∞	$-\infty$
(same, write, 1)	13	$+0.337$	∞	$-\infty$

Table 15: Post-hoc $K{=}1$ diagnostic on the validation fold ($q{=}0.05$, n required 19, $\rho_R{=}0.283$, $\lambda{=}1$). Six cells now obtain a finite radius, but every lower bound remains negative and deployment is still $0/32$. This is a diagnostic on data already seen, not a validated guarantee.

α	q	n req.	cells $\widehat{\beta} < \infty$	uses	mean U
0.10	0.025	39	$2/8$	$0/32$	0.496
0.15	0.037	26	$4/8$	$0/32$	0.496
0.20	0.050	19	$6/8$	$0/32$	0.496
0.30	0.075	13	$8/8$	$16/32$	0.960
0.40	0.100	9	$8/8$	$17/32$	0.975

Table 16: Validation fold, certificate behaviour against the declared joint error budget. Observed harm is 0 at every level that opens ($0/16$, $0/17$; exact 95% upper bounds 17.1%, 16.2%). The curve is reported in full; no level is selected.

U What the Coverage Radius Buys: Candidate-Set Ablations

On the deployment tasks the point-estimate rule ($\widehat{\tau} > 0$, no radius) matches TRACE-AVG on utility and observed harm while acting on all 96 requests rather than 32 (Table 2). That invites the obvious question: is the good outcome due to coverage calibration, or simply to structural features the point rule also sees? On this benchmark the answer is the latter, and the radius earns its place only under stress. We test that by ablating the blind-slice candidate set, recomputed from the committed caches at no additional cost.

Four of the eight authored types still name the correct tool, so deleting only the structural match changes nothing: the point rule selects *partial* instead and still reaches 1.000 at zero harm. The informative ablation removes helpful candidates, leaving harmful and inert ones. Reporting only the fully ablated endpoint would invite the reply that one adversarial candidate set proves little, so we report the whole curve. For a removal fraction $p \in \{0, 0.25, 0.5, 0.75, 1\}$ we drop a random p-fraction of the helpful types, average over 400 independent draws, and let every rule see the same draw. Helpful is defined on the slice mean, which is the definition the endpoint above uses, so $p{=}1$ reproduces it exactly.

Three readings matter. First, the separation is not an artifact of the endpoint: harm to the point rule rises monotonically well before every helpful candidate is gone. Second, TRACE-AVG does not achieve 0 harm by being inert, since at $p{=}0$ it acts on all 32 tasks at 1.000; it converts degrading candidate quality into abstention, which is the behaviour the radius is supposed to buy. Third, similarity is harmed at every level including $p{=}0$ and degrades fastest, so surface relevance is the wrong quantity throughout. Defining helpful per task instead of on the slice mean weakens the endpoint (point rule $2.0/32$ rather than 12.0) because tasks already at ceiling have no helpful candidate to remove; the ordering of the rules and TRACE-AVG's 0 harm at every p are unchanged. Two limits of the construction remain: the candidate set is authored, and each episode is a single tool call.

V Domain-Level Inference

The 96 practice tasks are phrasing and ambiguity variants over only 8 domains, so a task-clustered bootstrap treats surface variants as independent evidence and is anti-conservative for a claim about

Rule (blind slice, 32 tasks)	p=0	p=0.25	p=0.5	p=0.75	p=1
harmful uses (mean over 400 draws)					
similarity	2.0	1.6	2.9	11.2	18.0
point estimate only	0.0	0.0	0.2	2.9	**12.0**
TRACE-AVG (λ=0.25)	0.0	0.0	0.0	0.0	**0.0**
TRACE-CERT (λ=1)	0.0	0.0	0.0	0.0	0.0
uses					
similarity	32.0	32.0	32.0	32.0	32.0
point estimate only	32.0	32.0	32.0	32.0	32.0
TRACE-AVG (λ=0.25)	32.0	32.0	28.9	12.7	0.0
TRACE-CERT (λ=1)	0.0	0.0	0.0	0.0	0.0
mean utility (never memory is 0.479 at every p)					
similarity	0.938	0.951	0.862	0.389	0.000
point estimate only	1.000	1.000	0.994	0.882	0.479
TRACE-AVG (λ=0.25)	1.000	1.000	0.951	0.684	0.479

Table 17: Candidate-removal curve. Harm to the point-estimate rule is monotone in p and reaches $12/32$ at the fully ablated endpoint, while TRACE-AVG takes 0 harmful uses at every level and pays for it by withdrawing, its use rate falling $32 \to 0$ as the candidate set degrades. This is graceful degradation under candidate-set shift rather than the property of a single ablation. Recomputed from the committed caches; no model was called.

new domains. The domain is the correct resampling unit for that claim, and we report it here. All three analyses are recomputed from the committed caches and cost nothing.

Domain-clustered resampling does not widen the intervals. Table 18 resamples the 8 practice domains with replacement rather than the 96 tasks. Contrary to what one might expect, the intervals *narrow* rather than widen: between-domain variance is small relative to within-domain variance, because each domain contributes 12 tasks whose effects are nearly identical. The correct unit therefore does not soften the conclusion. The one type whose interval covers zero is *irrelevant* under both units, which is the inert class.

Memory type	$\widehat{\tau}$	**task-clustered**	**domain-clustered**
supportive	$+0.167$	$[+0.097, +0.243]$	$[+0.156, +0.177]$
structural match	$+0.167$	$[+0.097, +0.247]$	$[+0.156, +0.177]$
stale	$+0.167$	$[+0.097, +0.243]$	$[+0.156, +0.177]$
partial	$+0.163$	$[+0.094, +0.243]$	$[+0.153, +0.174]$
conflicting	$+0.139$	$[+0.066, +0.219]$	$[+0.111, +0.160]$
irrelevant	-0.014	$[-0.115, +0.083]$	$[-0.031, +0.003]$
lexical distractor	$\mathbf{-0.247}$	$[-0.330, -0.160]$	$[-0.330, -0.181]$
corrupted	$\mathbf{-0.420}$	$[-0.514, -0.326]$	$[-0.486, -0.358]$

Table 18: Paired transfer by memory type under two resampling units (2,000 resamples). Domain-clustered intervals are the appropriate unit for a new-domain claim and are *narrower* here, so the sign pattern is not an artefact of treating phrasing variants as independent.

Leave-one-domain-out. Holding out each practice domain in turn and refitting the cell means on the remaining seven, the sign of $\widehat{\tau}$ is preserved for *every* memory type in *all* 8 folds ($8/8$ for each of the eight types). No single domain drives the harmful sign of the lexical distractor or of corrupted memory.

Per-domain deployment behaviour. The headline policy result is an average over 96 held-out-domain tasks, which could in principle hide a domain where the gate does badly. It does not. Refitting $\widehat{\tau}$ and the envelope on practice only and applying the deployed rule (shortlist top-K=2 by train-fold $\widehat{\tau}$, use when the lower endpoint clears zero) at $\lambda^{\star}$=0.25, the behaviour is identical across all 8 held-out domains (campaign, database, device, domainreg, email, subscription, ticket, webhook; 12 tasks each): each domain records 4 uses, 8 deferrals, no fall-backs and 0 harmful

uses, at mean utility 1.000 against a never-memory baseline of 0.826. Summing gives the headline 32 uses and 64 deferrals over 96 tasks, so the aggregate is a uniform average rather than a strong domain covering a weak one. The uniformity is itself a property of the construction: domains share the read/write twin structure, so the cell a candidate lands in does not depend on which domain it came from.

W Sequential Organic Pilots: ALFWorld and a τ-bench-derived Environment

Both pilots were run to look for organic harmful transfer in a *sequential* environment, which neither the controlled benchmark (single tool call) nor BFCL (single call, at ceiling) can supply. Neither found harm. We record them because the second is the only setting we have that is simultaneously organic, multi-turn and *away from ceiling*, and so is the one that could most plausibly have exhibited the mechanism.

ALFWorld (68 tasks, text-only). On ALFWorld (Shridhar et al., 2021) the no-memory agent solved $68/68$, so the setting is at ceiling on the primary metric. Under paired injection of organically derived memories, binary success changed on $1/68$ tasks for the organic distractor and $0/68$ for the structural match, giving $\tau \approx -0.015$ and 0. We therefore also recorded a secondary efficiency-aware return U_{eff}, and two properties of that metric bound every claim built on it. First, when the baseline succeeds, $U_{0,\text{eff}} = 1$ and $U_{1,\text{eff}} = \min(1, s_0/s_1)$, so $U_{1,\text{eff}} - U_{0,\text{eff}} \leq 0$ by construction: a memory that solves the task in *fewer* steps cannot register a positive effect. U_{eff} is thus a one-sided slowdown diagnostic, not a utility on which *use* could ever be certified; a two-sided bounded return such as $U = \mathbf{1}\{\text{success}\} - \eta \min(s, S_{\max})/S_{\max}$ would be required for that. Second, the forced-use comparison is optimistic, because the binary rule breaks coverage-free ties toward *use* and breaking them toward fallback yields no degradation on this data. With those caveats, on U_{eff} the forced-use rule degraded $18/68$ tasks under a constructed coverage hole while the three-way rule deferred on all 68 and degraded none, and reallocating the cached paired rollouts (46 acquisition / 22 held-out) reduced the calibration radius from ∞ to 0.308, 0.267 and 0.145. The policy never moved off defer, so this is radius tightening, not decision resolution. The accompanying n_{eff} and acquisition-strategy analyses resample the same 68 episodes and define their own target from them, the implemented n_{eff} simplifies to an expected per-cell count, and all acquisition strategies ended at 100% defer with 0% resolved; we treat both as finite-cache sanity checks and draw no conclusion about the $n_{\text{eff}}^{-1/2}$ rate or about which acquisition rule is best.

A τ-bench-derived retail environment. This pilot, derived from τ-bench (Yao et al., 2025), is the one *not* at ceiling: the no-memory baseline solves roughly 64% of retail tasks on Haiku and 79% on Sonnet, so there was decision slack for a harmful memory to exploit, and the memories are real retrieved traces rather than notes we wrote. Organically retrieved distractors were nonetheless helpful rather than harmful ($\tau = +0.251$, $[+0.170, +0.330]$ on Haiku; $+0.188$, $[+0.106, +0.269]$ on Sonnet, $n = 45$ each), as were structural matches ($+0.284$ and $+0.208$), and only 2.2% and 4.4% of pairs fell below -0.1. This is the strongest evidence we have that the absent harmful class is not merely hidden by a ceiling: in a non-ceiling sequential environment with organically retrieved memories, reuse still helped. It is also the sharpest limit on the paper's scope, and we prefer to state it here rather than let a ceiling explanation stand in for it.

An injected-memory airline variant gave $\tau = -0.086$ ($[-0.200, +0.000]$, $n = 14$ of 50 tasks with a usable twin), with 28.6% of pairs below -0.1. We do *not* read it as mechanism evidence, because the agent *followed* the injected action in only 1 of 70 episodes: the score difference tracks the injection being delivered rather than obeyed. Both environments re-implement the tool loop and evaluator rather than running the official τ-bench harness, so we call this a τ-bench-*derived* environment and report no τ-bench benchmark number. Both pilots are analysed entirely from cached rollouts, so every number above is recomputed at no additional model cost.

X PREREGISTERED ORGANIC PILOT (APPWORLD, STEP 0)

The organic settings above each leave at least one opening: the AppWorld distractor is a playbook we wrote, the BFCL probes sit at ceiling, and the sequential pilots, one of which is genuinely off ceiling, retrieve from a small hand-assembled pool. This pilot closes the authorship opening as far as our budget allows, on a non-ceiling split, and it is reported here because its outcome is a bounded negative rather than a success.

Construction. AppWorld ships a reference solution for 147 of its tasks: a real, commented, multi-step API workflow written by the benchmark authors. The memory bank is those 147 workflows verbatim, so it costs no rollouts to build and contains no text of ours. Deployment tasks are the first ten *distinct task families* of the `test_normal` split, which our AppWorld probe (Appendix R) never touched; variants within a family are near-duplicates, so taking distinct families is what keeps ten tasks from collapsing to four independent items. Retrieval is the surface-similarity rule this paper argues is uninformative about transfer: rank the bank by token overlap with the task instruction and take the highest-ranked workflow whose primary app is *not* one the task needs. The harmful candidate is therefore selected by the mechanism we criticise rather than by us; what we control is retrieval, not content. Arms are *none* and the retrieved workflow, one rollout each, on the Sonnet executor with the turn cap, scorer and cell encoder carried over unchanged.

The gate and the stop rule, both fixed before any rollout. This is a Step-0 viability gate, not the experiment: it decides whether a full fold is worth running. **G1 (slack)** required mean $U(\text{none}) \leq 0.85$ with at least three tasks above zero, so that degradation is expressible at all and the split is not floored. **G2 (harm)** required at least 3 of 10 tasks to come out worse with the memory than without. Passing meant running the full fold; failing meant stopping rather than searching for a more favourable split, which would be selection on the outcome.

Outcome: G1 passed, G2 failed. Mean $U(\text{none}) = 0.605$ with $9/10$ tasks above zero, so the slice was genuinely off both ceiling and floor and the gate was a real test. The paired outcomes were **two negative, four positive and four tied**, against the three negatives G2 required. The mean paired difference was $+0.030$, that is, slightly *positive*. Per the stop rule we did not run the full fold, and we did not lower the threshold to two.

The two negatives are worth recording precisely because they are the first we have seen in an organic setting: one task fell from 0.800 to 0.100 and another from 0.600 to 0.400, against three prior organic probes in which *no* task was harmed at all ($0/57$ and $0/39$ on BFCL, and positive means on both sequential pilots).

Replication: neither negative reproduces. Because the gate ran one rollout per arm, we then collected an independent five-repetition block on the *same* ten tasks with the *same* retrieved memories, changing only the repetition index. This audits reproducibility; it does not revisit the gate verdict and does not estimate any base rate. Averaging the five repetitions within each arm before differencing, and treating the task family as the unit, both rep-0 negatives reverse sign: the $0.800 \to 0.100$ task moves to $+0.220$ (its five block differences are $0, +0.80, 0, +0.30, 0$, never negative) and the $0.600 \to 0.400$ task to $+0.080$. Across the ten families the block mean is $+0.111$ with a standard error of 0.053, and only one family has a negative block mean, that being a task whose rep-0 difference was $+0.600$. Per the pre-specified reading, both rep-0 drops are therefore reported as stochastic variation.

The sign instability is itself the useful observation: per-rollout variance on this slice is of the same magnitude as the observed effects, in both directions, which is exactly why a single rollout per arm cannot support a per-task transfer claim.

What may and may not be concluded. Each arm was run once, so the per-task numbers are single observed paired differences Δ, not estimates of the task-level transfer effect τ; no per-task effect, and no base rate of organic harm, is estimable from ten single rollouts. The defensible statement is that organic retrieval produced isolated negative paired outcomes, including one large observed degradation, but failed the prespecified reliability gate and remained slightly positive on average. This neither establishes an organic negative-transfer regime nor rules one out; it bounds what we

can claim, and it is why the paper says reliable harm is established only for authored competing memories. Cost: 20 rollouts, ≈$0.70.

Y MANIFEST-FROZEN SAME-APP DIRECTION PILOT

Appendix X failed partly because the injected workflow concerned an app the task did not need, so the agent could dismiss it. This pilot holds the app *fixed* and compares workflows classified as aligned or opposite-direction under a coarse action ontology. The workflow texts otherwise differ and are not exact action twins. Both are verbatim benchmark-authored workflows; we author nothing.

Design. Bank: the same 147 reference workflows. Targets: `test_normal` families untouched by our AppWorld probe and by Appendix X, one variant per family. An action ontology and an instruction parser were developed on train and dev *only* and validated against dev reference solutions (24 correct, 0 wrong, 6 abstentions on gold-labelled two-direction dev tasks), then applied unmodified. Eligibility requires exactly one two-direction app named, an unambiguous direction, a non-interrogative task, and both an aligned and an opposite-direction candidate in the bank; 13 of 46 families qualify, the rest failing on ambiguity (12), missing two-direction app (9), read-only phrasing (8) or multiple apps (4). Arms are *none*, *aligned* and *opposite-direction*, R=5, on the Sonnet executor. Candidates are chosen by token overlap with the instruction, ties by task id. The unit is the task family: the five repetitions are averaged within an arm *before* differencing.

Result: both memory arms help, and their difference is unresolved.

Contrast	Mean	Family-clustered 95% CI
aligned − none	+0.157	[+0.086, +0.231]
opposite-direction − none	+0.119	[+0.031, +0.214]
primary D = opposite − aligned	−0.038	[−0.107, +0.021]
sensitivity, venmo excluded (N=6)	−0.003	[−0.076, +0.060]

Both labelled workflow prefixes improve utility over no memory, and the primary contrast neither establishes harm nor excludes a modest one.

Four limitations that bound this pilot, all of which we checked rather than assumed. First, the injected memory was truncated to 1,400 characters, and on 10 of 13 aligned and 10 of 13 opposite-direction sources *no* action-bearing API name survives that cut, so in most runs the model never saw the conflicting action itself. Second, uptake must be read against the no-memory arm, not in absolute terms, and only endpoints whose API names survive in the presented prefix count: distinctive visible endpoints are reached in 19/65 aligned runs against 13/65 no-memory runs, but in only 11/65 opposite-direction runs against 9/65 no-memory runs, so opposite-direction-specific endpoint use barely rises above baseline. (An earlier version of this audit matched API *names* against URLs and reported zero uptake everywhere; names do not appear in paths, and the corrected mapping uses the benchmark's own OpenAPI operation identifiers.) Third, the ontology distinguishes only positive from negative direction, so pairs such as adding a Venmo friend against declining a payment request are same-app opposite-direction workflows rather than exact action twins. Fourth, a preregistration deviation: arm order was to be a seeded permutation per target, but the seed was derived from Python's salted `hash`, so 60 of 65 blocks ran in an order other than the one recorded in the frozen manifest. Arm order is therefore effectively arbitrary rather than reproducibly seeded; we report it because the manifest says otherwise. The frozen manifest's analysis hash consequently identifies the pre-audit script; the released script changes the process audit and archive-only loading, not the frozen tasks, arms, utility observations or primary utility formula.

The empirical-Bernstein radius of Corollary 1 is 1.53 at N=13 with range 2, so it is vacuous here and only the family-clustered bootstrap is informative.

What this does and does not support. Because action-bearing code was often truncated and opposite-direction endpoint use did not clearly rise above the no-memory baseline, this pilot does *not* establish that the executor detects and discounts conflicting procedures, and it does not establish

organic negative transfer. What it does support is narrower and still useful: verbatim workflow prefixes are broadly *procedurally helpful* on this benchmark under both coarse direction labels, which strengthens the scope boundary that the harm our controlled benchmark isolates is a property of an engineered competing memory rather than a general property of reused experience.

Z ORGANIC-MEMORY PROBE (BFCL SIMPLE)

To probe whether the effect survives when the memory *text* is not authored by us, we run an organic-memory-content diagnostic with oracle-controlled pair construction on BFCL V4 simple. Every memory is a verbatim real BFCL instance. For a target task solved by function f, we form: (i) **structural match**; another real task using the *same* f (different arguments/entity), shown as "a past request you handled successfully" with its ground-truth call; and (ii) **lexical distractor**, the real task with the highest surface-token overlap that uses a *different* function, again shown with its real solved call. No memory text is authored; benchmark function labels are used only to construct this diagnostic pairing and would not be available to a deployed retriever.

Split hygiene. We split task ids into a source bank and targets *before* constructing any memory: for each function with ≥ 2 instances the first id is reserved as a memory source (never a target) and the rest become targets; single-instance tasks join the source bank. Every memory supplied to any target is drawn *exclusively* from this source bank, and the target itself and all other targets are excluded, so no target's solved call can appear in another target's memory (no cross-task leakage). This yields a 370-task source bank and 30 targets (split 15 practice / 15 deploy by id); we run $\{$none, structural, distractor$\} \times R{=}3$ on Haiku (270 episodes, $\approx$\$0.3).

Result. Neither organic memory type has a reliably negative mean effect: structural match $\tau = +0.067$ $[+0.000, +0.167]$ and lexical distractor $\tau = +0.033$ $[+0.000, +0.100]$ (task-clustered bootstrap), both near zero to mildly positive. The regime is an extreme ceiling: never-memory utility 0.978, with most tasks already solved without memory and a *single* unambiguous candidate function, so the executor calls the correct function regardless of the memory. Deployment policies: never 0.978, always-structural 1.000, the average-value selector gate 1.000 (use $15/15$, $0/15$ observed harm), oracle 1.000.

Reading. This experiment isolates a useful scope boundary but does *not* by itself establish harmful transfer with organically generated memories, nor does it rule out authorship as a factor: the ceiling and single-candidate structure changed together with the memory source, so we report only that we do *not* observe systematic negative transfer in this unambiguous ceiling regime. Read alongside AppWorld it is a third scope datapoint: across our tested settings systematic negative transfer appears when a memory can redirect the executor among competing plausible actions, and this regime: like AppWorld: does not present that competition. It does address the "injected memory is artificial" concern to the extent that real reused traces in a ceiling regime behave as the structural account predicts (little leverage when the action is unambiguous).

Organic BFCL-Multiple and a direct CMI adaptation. BFCL-simple offers a single candidate function, so it cannot exhibit the competing-action structure the twin trap needs. We therefore ran a second organic probe on BFCL-Multiple, where each task offers 2–4 candidate functions. Split hygiene as above: a task is a target if one of its *competing* offered functions is the ground-truth of some other real task, which supplies an organic *distractor* memory (a verbatim real solved trace for the wrong-but-plausible action); the source bank is all non-target tasks, so no target's own trace is reused. This yields 39 targets (19 practice / 20 deploy). We additionally implement a direct adaptation of *CMI*'s (Srivastava, 2026) probe-based selection rule: for each candidate memory we run no-memory, with-memory, and perturbed-memory probes *at request time* and select the memory with positive utility (with-memory minus no-memory score) and non-negative stability (with-memory minus perturbed); this costs $2k$ extra executions per request (with-memory and perturbed probes for each of the k candidates), each requiring a request-time outcome score.

Result. The subset is again a ceiling ($37/39$ tasks solved without memory; never-memory 0.983), so the organic competing-function distractor is not reliably harmful ($\tau = +0.009$ $[+0.000, +0.026]$; it hurts $0/39$ tasks). Deployment: never 0.983, always-similarity 1.000 (0% harm), CMI 1.000 (0% harm, memory used on $1/20$), the average-value selector and the certified gate 0.983 (0% harm, both abstain), oracle 1.000. *Scorer disclosure:* CMI requires a request-time outcome score for each

Axis	CMI (Srivastava, 2026)	Ours
Intervention target	request-time answer utility	full-trajectory return
Extra decision probes / request	$(2k{+}1)R$ (51 here)	none
Deployment outcome scorer	needed (or surrogate)	not used
Practice→deployment transfer	not central	core problem
Average safe-improvement guarantee	–	the average-value selector
Pair-level uncertainty	heuristic stability	the certified gate (marginal)
Coverage-hole repair	–	active practice

Table 19: Our approach vs. the closest method (CMI). The difference is systemic, no request-time counterfactual probes and explicit guarantee semantics: rather than a benchmark-utility win.

of its no-memory, with-memory, and perturbed-memory probes; we provide it the benchmark's own execution evaluator as an oracle request-time scorer. This favors CMI and is unavailable in ordinary deployment; the same evaluator scores all three probe types. We use no request-time scorer. On the 20 deployment targets CMI solves one additional task ($20/20$ versus $19/20$ for ours); this small diagnostic subset is insufficient to resolve a reliable utility difference. We therefore do not claim statistical equivalence and do not claim a utility win: CMI yields one additional successful task on this ceiling subset but *no* detectable harm reduction (both 0%), at substantially higher request-time cost. The point is that on a public benchmark with competing candidates, (i) a real reused trace for a competing action is not automatically harmful when the executor is competent, and (ii) the closest method's request-time experimentation yields no harm-reduction advantage in this regime while costing more (cost accounting and break-even below).

Is the ceiling an artifact of a strong executor? A natural rescue is to weaken the executor until it becomes genuinely uncertain among the competing functions. We tested this directly by sweeping a ladder of smaller open tool-callers on the same BFCL-Multiple targets: Qwen3-coder-30B-A3B, Qwen3-32B, and Qwen3-next-80B-A3B: under the identical no-memory protocol. All three remain at ceiling (per-task never-memory scores ≈ 1.0; only Qwen3-32B dips to 0.5 on a single task), because the benchmark's requests are explicit enough that any competent tool-caller selects the correct function regardless of size. The ceiling is therefore a property of the *benchmark's explicit-request design*, not of executor strength: a smaller model does not manufacture the decision slack an unambiguous request removes. In our *organic-trace* probes, obtaining non-ceiling decision slack may require genuine ambiguity or multi-step compounding, which public single-turn tool-calling suites do not supply; this is *not* a general necessary condition for negative transfer, as the injected BFCL-Multiple result ($\tau = -0.087$ on explicit single-turn requests) demonstrates. Hence the informative next test is an organic *multi-step* twin trap rather than a weaker model on BFCL.